

Fair and Square: Cake-Cutting in Two Dimensions

Erel Segal-Halevi¹, Shmuel Nitzan², Avinatan Hassidim³ and Yonatan Aumann⁴

Bar-Ilan University, Ramat-Gan 5290002, Israel

Abstract

We consider the problem of fairly dividing a two-dimensional heterogeneous good, such as land or ad space in print and electronic media, among several agents with different utilities. Classic cake-cutting procedures either allocate each agent a collection of disconnected pieces, or assume that the cake is a one-dimensional interval. In practice, however, the two-dimensional shape of the allotted pieces may be of crucial importance. In particular, when building a house or designing an advertisement, squares are more usable than long and narrow rectangles. We thus introduce and study the problem of fair two-dimensional division wherein the allotted pieces must be of some restricted two-dimensional geometric shape(s). Adding this geometric constraint re-opens most questions and challenges related to cake-cutting. Indeed, even the most elementary fairness criterion - *proportionality* - can no longer be guaranteed. In this paper we thus examine the *level* of proportionality that *can* be guaranteed, providing both impossibility results (for proportionality that cannot be guaranteed) and division procedures (for proportionality that can be guaranteed). We consider cakes and pieces of various shapes, focusing primarily on shapes with a balanced aspect ratio such as squares.

JEL classification: D63

Keywords: Cake Cutting, Fair Division, Land Economics, Geometry

¹erelsgl@cs.biu.ac.il . Corresponding author.

²nitzans@biu.ac.il

³avinatan.hassidim@biu.ac.il

⁴aumann@cs.biu.ac.il

(a) Two disjoint rectangles worth $1/2$ (b) Two disjoint squares worth $1/4$ (c) No two disjoint squares worth more than $1/4$

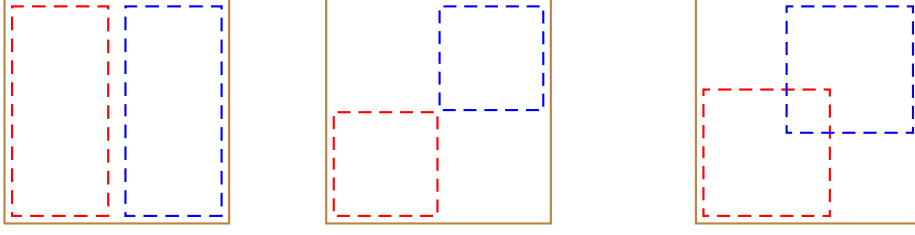


Figure 1: Dividing a square cake to two agents.

1. Introduction

Fair division of land has been an important issue since the dawn of history. One of the classic fair division procedures, “I cut - you choose”, is already alluded to in the Bible (Genesis 13) as a method for dividing land between two people. The modern study of this problem, commonly termed *cake cutting*, began in the 1940’s. The first challenge was conceptual - how should “fairness” be defined when the cake is heterogeneous and different people may assign different values to subsets of the cake? Steinhaus [1] introduced the elementary and most basic fairness requirement, now termed *proportionality*: each of the n agents should get a piece which he values as worth at least $1/n$ of the value of the entire cake. He also presented a procedure, suggested by Banach and Knaster, for proportionally dividing a cake among an arbitrary number of agents. Since then, many other desirable properties of cake partitions have been studied, including: envy-freeness [e.g. 2, 3, 4, 5], social welfare maximization [e.g. 6, 7, 8] and strategy-proofness [e.g. 9, 10, 11]. See the books by Brams and Taylor [3], Robertson and Webb [12], Barbanel [13], Brams [14] and a recent survey by Procaccia [15] for more information.

Many economists regard land division as an important application of division procedures [e.g. 16, 17, 18, 19, 20, 21, 22]). Hence, they note the importance of imposing some geometric constraints on the pieces allotted to the agents. The most well-studied constraint is *connectivity* - each agent should receive a single connected piece. The cake is usually assumed to be the one-dimensional interval $[0, 1]$ and the allotted pieces are sub-intervals [e.g. 23, 4, 24, 25]). This assumption is usually justified by the reasoning that higher-dimensional settings can always be projected onto one dimension, and hence fairness in one dimension implies fairness in higher dimensions.⁵ However, projecting back from the one dimension, the resulting two-dimensional plots are thin rectangular slivers, of little use in most practical applications; it is hard to build a house on a $10 \times 1,000$ meter plot even though its area is a full hectare, and a thin 0.1-inch wide advertisement space would ill-serve most advertisers regardless of its height.

We claim that the *two-dimensional shape* of the allotted piece is of prime importance. Hence, we seek divisions in which the allotted pieces must be of some restricted family of “usable” two-dimensional shapes, e.g. squares or polygons of balanced length/width ratio.

Adding a two-dimensional geometric constraint re-opens most questions and challenges related to cake-cutting. Indeed, even the elementary proportionality criterion can no longer be guaranteed.

Example 1.1. A homogeneous square land-estate has to be divided between two heirs. Each heir wants to use his share for building a house with as large an area as possible, so the utility of each

⁵In the words of Woodall [26]: “the cake is simply a compact interval which without loss of generality I shall take to be $[0, 1]$. If you find this thought unappetizing, by all means think of a three-dimensional cake. Each point P of division of my cake will then define a plane of division of your cake: namely, the plane through P orthogonal to $[0, 1]$ ”.

heir equals the area of the largest house that fits in his piece (see Figure 1). If the houses can be rectangular, then it is possible to give each heir $1/2$ of the total utility (a); if the houses must be square, it is possible to give each heir $1/4$ of the total utility (b) but impossible to give both heirs *more* than $1/4$ the total utility (c). In particular, when the allotted pieces must be square, a proportional division does not exist.⁶

This example invokes several questions. What happens when the land-estate is heterogeneous and each agent has a different utility function? Is it always possible to give each agent a 2-by-1 rectangle with a subjective value of at least $1/2$ the total value? Is it always possible to give each agent a square with a subjective value of at least $1/4$ the total value? Is it even possible to guarantee a positive fraction of the total value? If it is possible, what division procedures can be used? How does the answer change when there are more than two agents? Such questions are the topic of the present paper.

We use the term *proportionality* to describe the fraction that can be guaranteed to every agent. So when the shape of the pieces is unrestricted, the proportionality is always $1/n$, but when the shape is restricted, the proportionality might be lower. Naturally, the attainable proportionality depends on both the shape of the cake and the desired shape of the allotted pieces. For every combination of cake shape and piece shape, one can prove *impossibility results* (for proportionality levels that cannot be guaranteed) and *possibility results* (for the proportionality that can be guaranteed). While we examined many such combinations, the present paper focuses on several representative scenarios which, in our opinion, demonstrate the richness of the two-dimensional cake-cutting scene.

1.1. Walls and unbounded cakes

In Example 1.1, the two pieces had to be contained in the square cake. One can think of this situation as dividing a square island surrounded in all directions by sea, or a square land-estate surrounded by 4 walls: no land-plot can overlap the sea or cross a wall.

In practical situations, land-estates often have less than 4 walls. For example, consider a square land-estate that is bounded by sea to the west and north but opens to a desert to the east and south. Allocated land-plots may not flow over the sea shore, but they may flow over the borders to the desert.

Cakes with less than 4 walls can also be considered as unbounded cakes. For example, the above-mentioned land-estate with 2 walls can be considered a quarter-plane. The total value of the cake is assumed to be finite even when the cake is unbounded. When considering unbounded cakes, the pieces are allowed to be “generalized squares” with an infinite side-length. For example, when the cake is a quarter-plane (a square with 2 walls), we allow the pieces to be squares or quarter-planes. When the cake is a half-plane (a square with 1 wall), we also allow the pieces to be half-planes, etc. The terms “square with 2 walls” and “quarter-plane” are used interchangeably throughout the paper.

1.2. Fat objects

Intuitively, a piece of cake is usable if its lengths in all dimensions are balanced - it is not too long in one dimension and too short in another dimension. This intuition is captured by the concept of *fatness*, which we adapt from the computational geometry literature [e.g. 28, 29]:

⁶Berliant and Dunz [27] use a very similar example to prove the nonexistence of a competitive equilibrium when the pieces must be square.






Cake	Impossibility		Possibility		
	Squares	R -Fat rects ($R \geq 2$)	Squares	R -Fat rects ($R \geq 2$)	R -Fat polys ($R \geq 2$)
 4 walls (Square)	$1/(2n)$	$1/(2n - 1)$	$1/(4n - 4)$ <i>same:</i> $1/(2n)$	$1/(4n - 5)$ <i>same:</i> $1/(2n - 1)$	$1/(2n - 2)$
 3 walls	$1/(2n - 1)$		$1/(4n - 5)$, <i>same:</i> $1/(2n - 1)$		$1/(2n - 2)$
 2 walls (quarter-plane)	$1/(2n - 1)$		$1/(2n - 1)$		$1/(2n - 2)$
 1 wall (half-plane)	$1/(1.5n - 2)$		$1/(2n - 2)$		
 0 walls (plane)	$1/n$		$1/\max(2n - 4, n)$		

Table 1: Summary of results for square cakes: upper and lower bounds on the level of attainable proportionality.

Definition 1.1. A d -dimensional object is called R -fat, for $R \geq 1$, if it contains a d -dimensional cube c^- and is contained in a parallel d -dimensional cube c^+ , such that the ratio between the side-lengths of the cubes is at most R : $\text{len}(c^+)/\text{len}(c^-) \leq R$.

A two-dimensional cube is a square. So, for example, a square is 1-fat, a 10-by-20 rectangle is 2-fat, a right-angled isosceles triangle is 2-fat and a circle is $\sqrt{2}$ -fat.

Note that R is an upper bound, so if $R_2 \geq R_1$, every R_1 -fat piece is also R_2 -fat. So a square is also 2-fat, but a 10-by-20 rectangle is not 1-fat.

Our long-term research plan is to study various families of fat shapes. As a first step, we study the simplest fat shape, which is the square (hence the name of the paper). Despite its simplicity, it is still challenging. In the appendices, we generalize some of our results to other families of shapes such as fat-rectangles, square-pairs, multi-dimensional cubes, right-angled triangles, fat polygons and arbitrary fat pieces.

1.3. Results

1.3.1. Square cakes bounded or unbounded

We begin by studying situations in which the cake to be divided is a square bounded in zero or more sides. Table 1 summarizes our negative and positive results:

The **Impossibility** column shows upper bounds on the attainable proportionality. Each upper bound is proved by showing a specific scenario in which it is impossible to give all agents more than the mentioned fraction of their total value. The upper bound for a square with 4 walls and $n = 2$ is $1/(2n) = 1/4$, which is already seen in Example 1.1. For 0 walls our result is trivial: obviously it is impossible to guarantee more than $1/n$ to n agents.

The **Possibility** column shows our positive results. Each such result is proved constructively by an explicit division procedure that gives each agent at least the mentioned fraction of their total value.

The *same* result means that, when all agents have the same value measure, it is possible to guarantee each agent a larger fraction using a different division procedure (note that the impossibility results are valid whether the agents have the same or different value measures).

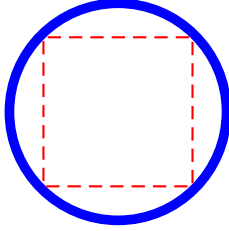


Figure 2: A circular cake where all value is near the perimeter. No positive value can be guaranteed to an agent who wants a square piece.

This difference illustrates an interesting economic implication of geometry. A common maxim in economics is that differences in preferences are desirable because they allow mutually-profitable trade. On the contrary, in two-dimensional cake-cutting we can guarantee better results when the preferences are identical.

Intuitively, one may think that allowing rectangles instead of just squares should considerably increase the attainable proportionality level. But this is not the case, as long as the pieces need to be fat. As seen in the table, both the possibility and impossibility results for fat rectangles are almost identical to the results for squares.

On the contrary, allowing non-rectangular polygons *does* make a difference. The possibility results in the “ R -fat polys” column are attained by a division procedure that gives each agent a 2-fat polygon in which all angles are multiples of 45 degrees (such as a right-angled isosceles triangle). The proportionality level is better than for R -fat rectangles and it is valid whether the agents have identical or different value measures. Moreover, the division procedure is much simpler. It is easier to divide a cake fairly when 45-degree cuts are allowed; this may explain why practical land allocation maps usually contain more than just rectangles.

Note that for $n = 2$, the proportionality levels in our possibility results are equal to the impossibility results. Note also that for a quarter-plane cake and rectangular pieces, the guaranteed proportionality is equal to the impossibility result for every n . This means that in these cases, our procedures are optimal in their worst-case guarantee. For a cake with 3 and 4 walls, the guaranteed proportionality for agents with the same value measure is optimal. In the other cases, there is a multiplicative gap of at most 2 between the possibility and the impossibility result.

1.3.2. Bounded cakes of any shape

While some states in the USA are rectangular (e.g. Colorado or Wyoming), most land-estates have irregular shapes. In such cases, it may be impossible to guarantee any positive proportionality. For example, consider Robinson Crusoe arriving at a circular island. Assume that Robinson’s value measure is such that all value is concentrated in a very thin strip along the shore, as in Figure 2. The value contained in any single square might be arbitrarily small. Clearly, no division procedure for n agents can guarantee a better fraction of the total value.

Therefore, for arbitrary cakes we use a *relative* rather than absolute fairness measure. For each agent, we calculate the maximum value that this agent can attain in a square piece if he doesn’t need to share the cake with other agents. We guarantee the agent a certain fraction of this value, rather than a certain fraction of the entire cake value. This fairness criterion is similar to the *uniform preference externalities* criterion suggested by Moulin [30]. Similar criteria have been recently studied in the context of indivisible item assignment [31, 32, 33].

Table 2 summarizes our bounds on relative proportionality. The impossibility results follow trivially from those for square cakes. The possibility results require new division procedures. They




Pieces		Impossibility	Possibility
Parallel squares		$1/(2n)$	$1/(8n - 6)$ <i>same:</i> $1/(2n)$
General squares		$1/(2n)$	$1/(16n - 14)$ <i>same:</i> $1/(2n)$
Parallel R -fat rectangles		$1/(2n - 1)$	$1/([4R + 4][n - 1] + 2)$ <i>same:</i> $1/(2n)$

Table 2: Summary of results for arbitrary compact cakes: bounds on the level of attainable relative proportionality.

are valid for any cake that is a compact (closed and bounded) subset of the plane. Note that the guarantees are better when the pieces are required to be axis-parallel. This is in accordance with the common practice in urban planning, in which axis-parallel plots are preferred. ⁷

1.4. Techniques

Introducing the geometric constraints required us to develop new proof and procedural techniques, making use of geometric notions.

Our main tool is a geometric generalization of the standard *mark* and *eval* queries used in the cake-cutting literature [e.g. 34, 12, 35]. A key geometric concept is *minimum square covering* - the minimum number of squares required to cover a given region.

Several different kinds of procedures are used. For square cakes with 4 and 3 walls, the procedures recursively divide the cake to sub-cakes using *guillotine cuts* - axis-parallel cuts running from one end of a piece to the opposite end (guillotine cuts are common in the industry e.g. for cutting plates of glass). For square cakes with 2 walls, the procedure carves the squares from the walls of the cake, one square at a time. For square cakes with 1 or 0 walls, the procedures combine guillotine cuts and carving. For arbitrary compact cakes, the procedure uses an algorithm for finding *maximum independent sets* - maximum collections of objects which do not overlap (similar procedures are used e.g. by mapping software in order to select which location labels to display on a map).

We note that besides fair division problems, geometric methods have been used in many other economics problems,⁸ such as voting [36], trade theory and growth theory [e.g. 37], tax burdens [38], social choice [39], mechanism design [40], public good/bad allocation [e.g. 41, 42], utility theory [43] and general economics models [44].

1.5. Paper structure

The remainder of the paper is structured as follows. The introduction section is concluded by reviewing some related research. The model is formally presented in Section 2. Impossibility results are proved in Section 3. Division procedures are described in Section 4. The main paper body focuses on results for square pieces; results for pieces of other shapes are presented in the appendices. Section 5 discusses several directions for future research.

⁷In Subsection 1.3.1 the results are the same for parallel and general squares.

⁸We are thankful to Steven Landsburg, Michael Greinecker, Kenny LJ, Alecos Papadopoulos, B Kay and Martin van der Linden for contributing these references in economics.stackexchange.com website (<http://economics.stackexchange.com/q/6254/385>).

1.6. Related work

The most prominent geometric constraint in cake-cutting is one-dimensional: the pieces must be contiguous intervals. Several authors studied a circular cake [45, 46, 47], but it is still a one-dimensional circle and the pieces are one-dimensional arcs.

Only few cake-cutting papers explicitly consider a two-dimensional cake. Two of them discuss the problem of dividing a disputed territory between several bordering countries, with the constraint that each country should get a piece adjacent to its current border: Hill [48] proved that such a partition exists and Beck [49] complemented this proof with a division procedure. Iyer and Huhns [50] describe a procedure that asks each of the n agents to draw n disjoint rectangles on the map of the two-dimensional cake. These rectangles are supposed to represent the “desired areas” of the agent. The procedure tries to give each agent one of his n desired areas. However, the procedure does not succeed unless each rectangle proposed by an individual intersects at most one other rectangle drawn by any other agent. If even a single rectangle of Alice intersects two rectangles of George (for example), then the procedure fails and no agent receives any piece.

Ichiishi and Idzik [51], Dall’Aglia and Maccheroni [20] acknowledge the importance of having nicely-shaped pieces in resolving land disputes. They prove that, if the cake is a simplex in any number of dimensions, then there exists an envy-free and proportional partition of the cake into polytopes. However, this proof is purely existential when the cake has two or more dimensions. Additionally, there are no restrictions on the fatness of the allocated polytopes and apparently these can be arbitrarily thin triangles. Berliant and Dunz [27] studies the existence of competitive equilibrium with utility functions that may depend on geometric shape; their nonwasteful-partitions-assumption explicitly excludes fat shapes such as squares. Devulapalli [52] studies a two-dimensional division problem in which the geometric constraints are connectivity, simple-connectivity and convexity.

In our model (see Section 2), the utility functions depend on geometry, which makes them non-additive. They are not even subadditive like in the models of Maccheroni and Marinacci [53], Dall’Aglia and Maccheroni [54]. Previous papers about cake-cutting with non-additive utilities can be roughly divided to two kinds: some [27, 55, 20, 56] handle general non-additive utilities but provide only pure existence results. Others [4, 57, 58] provide constructive division procedures but only for a 1-dimensional cake. Our approach is a middle ground between these extremes. Our utility functions are more general than the 1-dimensional model but less general than the arbitrary utility model; for this class of utility functions, we provide both existence results and constructive division procedures.

2. Model and Terminology

The *cake* C is a Borel subset of the two-dimensional Euclidean plane \mathbb{R}^2 . Usually C is a polygonal domain. *Pieces* are Borel subsets of \mathbb{R}^2 . *Pieces of* C are Borel subsets of C .

There is a pre-specified family S of pieces which are considered *usable*. An *S-piece* is an element of S .

In this paper we focus on the effect of geometric shape, so all the families we study are closed under translation, scaling and reflection (i.e, if an *S-piece* is translated, scaled and/or reflected, the result is also an *S-piece*). Most of the families are also closed under rotations. In the main body of this paper, S is usually the family of squares. More exotic families of usable pieces are examined in Appendix D.

C has to be divided among $n \geq 1$ *agents*. Each agent $i \in \{1, \dots, n\}$ has a value-density function v_i , which is an integrable, non-negative and bounded function on C . The *value* of a piece X to

agent i is marked by $V_i(X)$ and it is the integral of the value-density over the piece:

$$V_i(X) = \iint_X v_i(x, y) dx dy$$

When C is unbounded, we assume that the v_i are nonzero only in a bounded subset of C . Hence the V_i are always finite. The V_i are measures and are *absolutely continuous with respect to the Lebesgue measure* (or just *continuous* for short), i.e., any piece with an area of 0 has a value of 0. Hence, the value of a piece is the same whether or not it contains its boundary.

Based on V_i and S we define the following shape-based *utility* function, which assigns to each piece $X \subseteq C$ the value of the most valuable usable piece contained in X :

$$V_i^S(X) = \sup_{s \in S \text{ and } s \subseteq X} V_i(s)$$

For example, suppose S is the family of squares. If Alice wants to build a square house but gets a piece X that is not square, then she builds her house on the most valuable square contained in X . Hence her utility is the value of that most valuable square. We use the term *value* to refer to the (additive) measure V and the term *utility* to refer to the function V^S . Note that V^S is, in general, not a measure since it is not additive (it is not even subadditive). Hence, classic cake-cutting results, which require additivity, are not applicable.

An S -allocation is a vector of n S -pieces (X_1, \dots, X_n) , one piece per agent, such that $X_i \subseteq C$ and the X_i are pairwise-disjoint. Some parts of the cake may remain unallocated (free disposal is assumed). Since X_i is an S -piece, $V^S(X_i) = V(X_i)$. However, in general $V^S(C) < V(C)$.

The fairness of an allocation is determined by the agents' *normalized* values. Values can be normalized in two ways: either divide them by the *absolute* cake value for the agent and get $V_i(X_i)/V_i(C)$, or divide them by the *relative* cake utility for the agent and get $V_i(X_i)/V_i^S(C)$. Throughout the paper absolute normalization is used, except in Subsection 4.5 where relative normalization is used.

An allocation is called *proportional* if the normalized value of every agent is at least $1/n$. Example 1.1 shows that a proportional allocation is not always attainable (whether absolute or relative normalization is used). Hence, we define:

Definition 2.1. (*Absolute proportionality*) For a cake C , a family of usable pieces S and an integer $n \geq 1$:

(a) The *proportionality level* of C , S and n , marked $\text{Prop}(C, S, n)$, is the largest fraction $f \in [0, 1]$ such that, for every set of n value measures (V_1, \dots, V_n) , there exists an S -allocation (X_1, \dots, X_n) for which $\forall i : V_i(X_i)/V_i(C) \geq f$.⁹

(b) The *same-value proportionality level* of C , S and n , marked $\text{PropSame}(C, S, n)$, is the largest fraction $f \in [0, 1]$ such that, for every single value measure V , there exists an S -allocation (X_1, \dots, X_n) for which $\forall i : V(X_i)/V(C) \geq f$.

The analogous definition for relative proportionality is given in Subsection 4.5.

Obviously, for every C , S and n : $\text{Prop}(C, S, n) \leq \text{PropSame}(C, S, n) \leq 1/n$.

Applying this notation, classic cake-cutting results [e.g. 1] imply that for every cake C

$$\text{Prop}(C, \text{All}, n) = \text{PropSame}(C, \text{All}, n) = 1/n$$

⁹Shortly: $\text{Prop}(C, S, n) = \inf_V \sup_X \min_i V_i(X_i)/V_i(C)$, where the infimum is on all combinations of n value measures (V_1, \dots, V_n) , the supremum is on all S -allocations (X_1, \dots, X_n) and the minimum is on all agents $i \in \{1, \dots, n\}$.

where "All" is the collection of all Borel subsets of \mathbb{R}^2 . That is: when all pieces are usable, for every cake C and every combination of n continuous value measures there is a division in which each agent receives a utility of $1/n$, which is the best that can be guaranteed. One-dimensional procedures with contiguous pieces prove that $\text{Prop}(\text{Interval}, \text{intervals}, n) = 1/n$ and when translated to two dimensions they yield:

$$\text{Prop}(\text{Rectangle}, \text{rectangles}, n) = \text{PropSame}(\text{Rectangle}, \text{Rectangles}, n) = 1/n$$

However, these procedures do not consider constraints that are two-dimensional in nature, such as square pieces. Such two-dimensional constraints are the focus of the present paper.

Our challenge in the rest of this paper will be to establish bounds on $\text{Prop}(C, S, n)$ and $\text{PropSame}(C, S, n)$ for various cake shapes C and piece families S . Two types of bounds are provided:

- Impossibility results (upper bounds), of the form $\text{Prop}(C, S, n) \leq f(n)$ where $f(n) \in [0, 1]$, are proved by showing a set of n value measures on C , such that in any S -allocation, the value of one or more agents is *at most* $f(n)$. Such bounds are established in Section 3.
- Positive results (lower bounds), of the form $\text{Prop}(C, S, n) \geq g(n)$ where $g(n) \in [0, 1]$, are proved by describing a division procedure which finds, for every set of n value measures on C , an S -allocation in which the value of every agent is *at least* $g(n)$. Such bounds are established in Section 4.

2.1. The procedural model

Our division procedures are *query-based*. This means that they do not require full knowledge of the entire value measures. Rather, they send queries to the agents and determine the division according to the agents' replies. We elaborate more on the query model in Subsection 4.2.

For the sake of simplicity, our division procedures are presented as if all agents reply according to their true value functions. However, the guarantees of our procedures are stronger: they are valid for any *single* agent replying according to his own value function, regardless of what the other agents do. This is the common practice in the cake-cutting world.¹⁰

On the other hand, our procedures are not *dominant-strategy truthful* since an agent who knows the other agents' value functions may gain from mis-representing his own value function. Designing truthful cake-cutting mechanisms is known to be a difficult problem even with a 1-dimensional cake[10], and we leave it for future work.

3. Impossibility Results

Our impossibility results are based on the following scenario.

- The cake C is a desert containing k water-pools.
- Each pool is a square with side-length $\epsilon > 0$ (a very small constant).
- There are n agents with the same value measure: the value of a piece is proportional to the total amount of water in the piece.

¹⁰In the words of Steinhaus [1]: "The greed, the ignorance, and the envy of other partners cannot deprive him of the part due to him in his estimation; he has only to keep to the methods described above. Even a conspiracy of all other partners with the only aim to wrong him, even against their own interests, could not damage him."

a. $\text{Prop}(C, \text{Squares}, 2) \leq 1/3$ b. $\text{Prop}(C, \text{Squares}, 3) \leq 1/5$

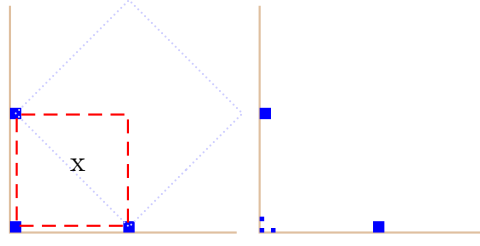


Figure 3: Impossibility results in a quarter-plane cake.

a. $\text{Prop}(C, \text{Squares}, 2) \leq 1/4$ b. $\text{Prop}(C, \text{Squares}, 3) \leq 1/6$

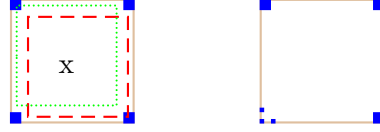


Figure 4: Impossibility results in a square cake.

- Thus, a piece that touches at most one pool has a value of at most $1/k$.

The k pools are arranged such that at most $n - 1$ disjoint S -pieces touch two or more pools. Hence, in every allocation of n S -pieces to the agents, at least one agent receives a plot which touches at most a single pool. This agent receives a value of at most $1/k$. This implies that: $\text{PropSame}(C, S, n) \leq 1/k$, which implies that $\text{Prop}(C, S, n) \leq 1/k$.

3.1. Impossibility results for two, three and four walls

We start with two results for two agents. Then we extend them to n agents.

Claim 3.1.

$$\text{PropSame}(\text{Quarter plane}, \text{Squares}, 2) \leq 1/3$$

Proof. Let I_3 be the set of 3 pools shown in Figure 3/a, where the bottom-left corners of the pools are in $(0,0)$, $(10,0)$, $(0,10)$. Every square in C touching two pools of I_3 must contain e.g. the point $(6,6)$ in its interior (marked by x in the figure). Hence, every two squares touching two pools of I_3 must overlap. Hence, at in any set of two disjoint squares, at least one of them touches at most one pool of I_3 . The agent receiving this square has a value of at most $1/3$. \square

Claim 3.2.

$$\text{PropSame}(\text{Square}, \text{Squares}, 2) \leq 1/4$$

Proof. Analogous to the previous claim. See Figure 4/a. \square

For the impossibility results for $n > 2$, we create sets of pools by deflating smaller sets into pools of other sets.

As an example, consider the arrangement I_3 from the proof of Claim 3.1. Suppose the entire plane is deflated (shrunk) towards the origin. If the deflation factor is sufficiently large, all three pools of the deflated I_3 are contained in $[0, \epsilon] \times [0, \epsilon]$, which is a pool of the original I_3 . The cake itself (the quarter-plane) is not changed by the deflation. By adding the other two pools of I_3 , namely $(10,0)$ and $(0,10)$, we get a larger pool set, I_5 , which is depicted in Figure 3/b. At most one square touches two pools in the deflated I_3 ; at most one square touches a pool from I_3 and one

of the new pools; hence, at most two squares touch two pools in I_5 . This proves that $\text{PropSame}(\text{Quarter plane}, \text{Squares}, 3) \leq 1/5$. We now generalize this idea to arbitrary n .

Claim 3.3. For every $n \geq 1$:

$$\text{PropSame}(\text{Quarter plane}, \text{Squares}, n) \leq \frac{1}{2n-1}$$

Proof. ¹¹It is sufficient to prove that for every n there is an arrangement of $2n-1$ pools in C such that at most $n-1$ disjoint squares touch two or more pools. The proof is by induction on n . The base case $n=1$ is trivial (and the case $n=2$ is Claim 3.1). For $n > 2$, assume there is such an arrangement with $2(n-1)-1$ pools. Deflate the entire arrangement towards the origin until it is contained in $[0, \epsilon] \times [0, \epsilon]$, where $\epsilon > 0$ is a sufficiently small constant. By the induction assumption, at most $n-2$ disjoint squares touch at least two of these deflated pools.

Add two new pools with side-length ϵ cornered at $(10, 0)$ and $(0, 10)$. We now have an arrangement of $2n-1$ pools. By our proof to Claim 3.1, every square touching a new pool and another pool (either new or old), must contain e.g. the point $(6, 6)$ in its interior, so every two such squares must overlap. Hence the total number of disjoint squares is at most $(n-2) + 1 = n-1$. \square

The upper bound for two walls is also trivially true when the cake is a square with three walls, since adding walls cannot increase the proportionality:

$$\text{PropSame}(\text{Square with 3 walls}, \text{Squares}, n) \leq \frac{1}{2n-1}$$

The bound also holds for a square with 4 walls, but in this case a slightly tighter bound is true:

Claim 3.4. For every $n \geq 2$,

$$\text{PropSame}(\text{Square with 4 walls}, \text{Squares}, n) \leq \frac{1}{2n}$$

Proof. W.l.o.g. assume C is the square $[0, 10+\epsilon] \times [0, 10+\epsilon]$. Create the arrangement of $2(n-1)-1$ pools from the induction step of Claim 3.3. Deflate it into $[0, \epsilon] \times [0, \epsilon]$. Add *three* new pools with side-length ϵ cornered at $(10, 0)$, $(0, 10)$ and $(10, 10)$, as in Figure 4/b. At most $n-2$ squares touch a deflated pool. Every square in C touching a new pool and another pool must contain $(5, 5)$ in its interior. Hence the total number of disjoint squares is still at most $(n-2) + 1 = n-1$, but now the total number of pools is $2n$. \square

3.2. Impossibility results for one wall

The results for one wall are analogous to the results for two walls.

Claim 3.5.

$$\text{PropSame}(\text{Half plane}, \text{Squares}, 3) \leq 1/4$$

Proof. Let I_4 be the set of 4 pools shown in Figure 5/a. Assume the side-length of each pool is $\epsilon \leq 0.01$ and that their bottom-left corner is in $(0, 0)$, $(5, 0)$, $(0, 10)$, $(-5, 0)$. We now examine the squares in C that touch two pools:

- Every square touching $(5, 0)$ and another pool must contain the point $x(4, 4.5)$ in its interior.

¹¹We are grateful to Boris Bukh for the idea underlying this proof.

$$\text{a. } \text{PropSame}(C, \text{Squares}, 3) \leq 1/4$$

$$\text{b. } \text{Prop}(C, 5, \text{Squares}) \leq 1/7$$

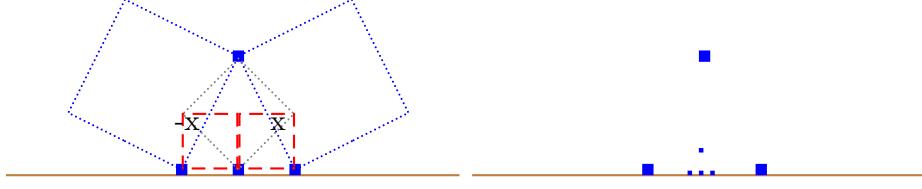


Figure 5: Impossibility result for 3 agents on a half-plane. See Claim 3.5

- Every square touching $(-5, 0)$ and another pool must contain the point $-x$ $(-4, 4.5)$.
- Every square touching $(0, 0)$ and another pool must touch either $(4, 4.5)$ or $(-4, 4.5)$ or both.

Hence, in every set of three squares touching two pools, at least two must overlap. Hence, in any set of three disjoint squares, at least one of them touches at most one pool. The agent receiving this square has a value of at most $1/4$. \square

Claim 3.6. For every odd $n \geq 3$:

$$\text{PropSame}(\text{Half plane}, \text{Squares}, n) \leq \frac{1}{1.5n - 0.5}$$

Proof. The proof is analogous to Claim 3. With each induction step, *three* new pools are added but only *two* new disjoint squares can be added. Hence the coefficient of n is $3/2$. The details are omitted. \square

Figure 5/b illustrates the construction for $n = 5$, having with 7 pools.

The bound of Claim 3.6 is obviously also valid when we put $n + 1$ instead of n in the left-hand side. In this case the right-hand side becomes $1.5n - 2$. Hence we get:

Claim 3.7. For every $n \geq 2$:

$$\text{PropSame}(\text{Half plane}, \text{Squares}, n) \leq \frac{1}{1.5n - 2}$$

Our impossibility results are almost identical when the usable pieces are R -fat squares. See Appendix D.1.

4. Possibility Results

4.1. Cover numbers

A key geometric concept in our possibility results is the *covering number* of the cake.

Definition 4.1. Let C be a cake and S a family of pieces.

- An S -cover of C is a set of S -pieces whose union equals C .
- The S -cover number of C , $\text{CoverNum}(C, S)$, is the minimum cardinality of an S -cover of C .

Some examples are depicted in Figure 6. The cover number can help Robinson Crusoe attain a valuable piece of an island:

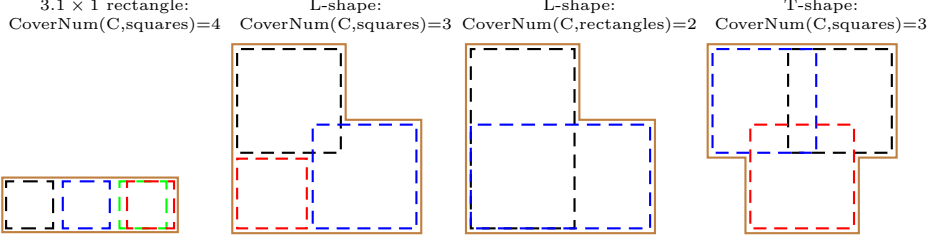


Figure 6: Cover numbers of various polygons.

Lemma 4.1. (*Covering Lemma*) *For every cake C and family S :*

$$Prop(C, S, n = 1) \geq \frac{1}{CoverNum(C, S)}$$

Proof. Let $k = CoverNum(C, S)$ and let $\{C_1, \dots, C_k\}$ be an S -cover of C . By definition $\cup_{j=1}^k C_j = C$. By additivity $\sum_{j=1}^k V(C_j) \geq V(C)$. By the pigeonhole principle there is j for which $V(C_j) \geq V(C)/k$. This C_j , which is an S -piece, gives the single agent a utility of at least $1/k$ of the total. \square

4.2. Queries

We now present division procedures that use two types of queries. These are generalizations of the *mark* and *eval* queries used in classic cake-cutting procedures [e.g. 12, 35].

In a **mark query**, the procedure specifies a value v , and each agent has to mark a piece of the cake with a subjective value of exactly v . Importantly, the procedure restrict the marked pieces such that they are totally ordered by containment (i.e. each marked piece either contains or is contained in every other marked piece).

In an **eval query**, the procedure partitions the cake C into $m \geq 2$ disjoint “rooms” C_j , with $\cup_{j=1}^m C_j = C$. Each agent i reveals his subjective value for each of the rooms, $V_i(C_j)$.

Example 4.1. A square cake has to be divided between 2 agents who want square pieces. In Example 1.1 we saw that the maximum utility that can be guaranteed to both agents is $1/4$ of the total value. We now present a division procedure that guarantees this utility.

For ease of presentation, we assume that the total cake value for each agent is 4 and present a procedure that allocates each agent a square with a value of at least 1. From now on we apply a similar convention: the total cake value is always scaled such that the value guarantee for each agent is 1.

The procedure is presented below. The reader may find it easier to follow the flowchart in Figure 7.

a. **Eval query:** partition the cake to a 2×2 grid and ask each agent to evaluate the $m = 4$ quarters. For each agent, choose one quarter which has a value of at least 1 (by the pigeonhole principle, there must be such a quarter).

b. If the chosen quarters are different, then give each agent his chosen quarter (and discard the rest of the cake). Each agent now holds a square with a subjective value of at least 1 and we are done. Otherwise, the chosen quarter is the same and we know that both agents value it as at least 1; proceed to the next query.

c. **Mark query:** ask each agent to draw, *inside* the chosen quarter and adjacent to the corner of C , a square with value *exactly* 1. This is always possible because the value measure is continuous.

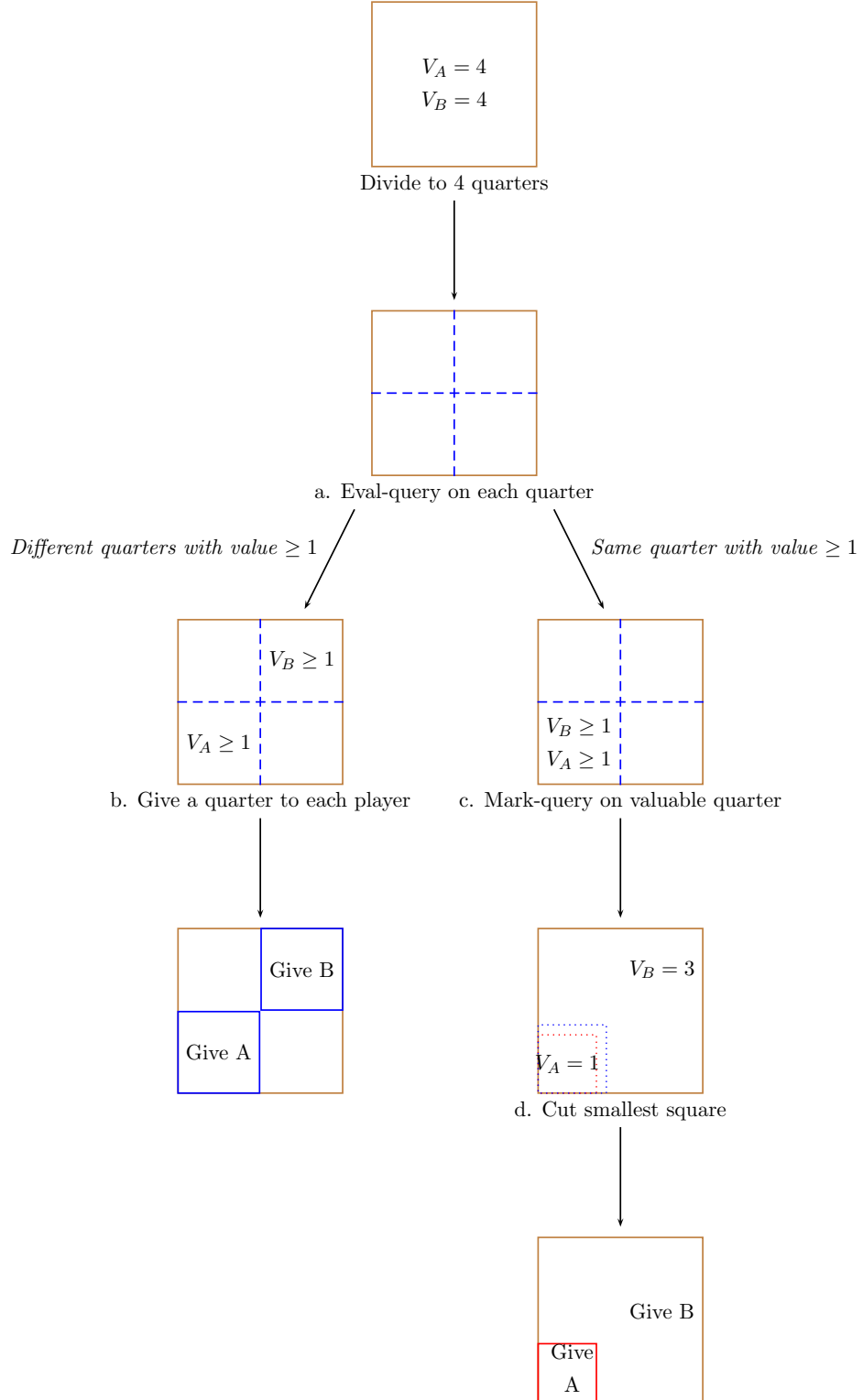


Figure 7: Dividing a square to two agents that want square pieces.

LEGEND FOR FLOWCHARTS:

- Dashed lines are lines drawn by the procedure before issuing *eval* queries, showing the pieces that the agents should evaluate.
- Dotted lines are lines drawn by the agents as responses to *mark* queries.
- Solid lines are cuts made by the procedure based on the answers to queries.

Note that the marked squares are totally ordered by containment - each square either contains or is contained in the other square.

d. Cut the *smaller* of the two squares and give it to the agent who drew it (breaking ties arbitrarily). This agent now holds a square which he values as exactly 1. The other agent values the allocated square as at most 1. By additivity, she thus values the remainder as at least 3. The remainder is an L-shape similar to the one in Figure 6. Its cover number is 3, so by the Covering Lemma it contains a square with a utility of at least 1; give that square to the remaining agent and finish.

Combining the lower bound proved by the procedure with the upper bound proved by Claim 3.2 gives a tight result for 2 agents:

$$\text{Prop}(\text{Square}, \text{Squares}, n = 2) = 1/4$$

Our procedures for n agents typically use the mark and eval queries in the following way.

After a **Mark query**, the procedure selects the smallest marked piece (which is contained in all other pieces), gives it to the agent who marked this piece, and divides the remainder recursively among the remaining $n - 1$ agents. As an example, suppose the cake C is a rectangle with a total value of n , and S is the family of rectangles. The cake can be divided using the following mark query, which is just a variation of [1]: “*mark a rectangle with a value of 1 whose left side coincides with the left side of C* ”. The rectangles differ only in their right side so they are totally ordered by containment, as required. The procedure then gives the smallest rectangle to its agent and divides the remainder recursively among $n - 1$ agents. Since the remaining $n - 1$ agents value the remainder as at least $n - 1$, by induction the final utility of each agent is at least 1.

After an **Eval query** on the rooms $\{C_j\}_{j=1}^m$, the procedure needs to partition the agents to groups $\{G_j\}_{j=1}^m$, and divide each room C_j recursively to the agents in group G_j . In doing so, the procedure must make sure that for every agent in group G_j , the value of room C_j is sufficiently large to guarantee that his final utility will be at least 1. This is done in two steps.

a. *Calculating the partner-numbers*: the procedure calculates, for each agent i and room j , a non-negative integer $P_{i,j}$, which is a function of $V_i(C_j)$. P stands for “partners” and represents the maximum number of agents, including i himself, with whom agent i can share room C_j while still being guaranteed a value of at least 1. As an example, suppose the rooms are rectangular and S is the family of rectangles. Then $P_{i,j} = \lfloor V_i(C_j) \rfloor$. E.g, if agent i values room j as 3.9 then $P_{i,j} = 3$, since if room j is divided among 3 agents, agent i is guaranteed a value of at least 1, but if it is divided among 4 agents, agent i might get less than 1.

b. *Room Partitioning*: Using the partner-numbers $P_{i,j}$, the procedure applies the following **Room Partition Algorithm (RPA)**, for which we are thankful to Christopher Culter [59].

INPUT: $N = \{1, \dots, n\}$ (a set of *agents*) and $M = \{1, \dots, m\}$ (a set of *rooms*).

For each agent $i \in N$, a set of m non-negative *partner-numbers* $P_{i,j}$ (with $j \in M$) such that $\sum_{j \in M} P_{i,j} \geq n$ (in words: for every agent, the sum of the partner numbers is at least the total number of agents).

OUTPUT: A partitioning of N to m disjoint groups G_j (some possibly empty), one group per room, such that: For every $i \in N, j \in M$: if $i \in G_j$ then $P_{i,j} \geq |G_j|$ (in words: for every agent in group j , the partner-number for room j is at least as large as the number of agents in group j).

ALGORITHM:

Room / agent	#1	#2	#3	#4
left	$V = 0.8 \rightarrow P = 0$	$V = 2.4 \rightarrow P = 2$	$V = 3.1 \rightarrow P = 3$	$V = 4 \rightarrow P = 4$
right	$V = 4.2 \rightarrow P = 4$	$V = 2.6 \rightarrow P = 2$	$V = 1.9 \rightarrow P = 1$	$V = 1 \rightarrow P = 1$
total	$\sum V = 5; \sum P = 4$	$\sum V = 5; \sum P = 4$	$\sum V = 5; \sum P = 4$	$\sum V = 5; \sum P > 4$

Table 3: Data for Room Partition Algorithm example: dividing two rectangular rooms among 4 agents.

- Initialize $G_m := \emptyset$.
- Order the agents in decreasing order of their $P_{i,m}$, breaking ties arbitrarily.
- While $P_{i,m} > |G_m|$: Let $G_m := G_m \cup \{i\}$.
- If $m > 1$, recursively call the procedure with agents $N \setminus G_m$ and rooms $M \setminus \{m\}$.

Proof. The RPA starts by filling the last room (m). It adds agents to G_m in a decreasing order of their partner-number for room m , until it is no longer possible to add another agent because the number of agents is larger than the next agent's partner-number. When the while-loop terminates, all agents in G_m have $P_{i,m} \geq |G_m|$ and all agents not in G_m have $P_{i,m} \leq |G_m|$. Thanks to the pre-condition $\sum_{j \in M} P_{i,j} \geq n$, for every agent i not in G_m : $\sum_{j \in M \setminus \{m\}} P_{i,j} \geq (n - |G_m|)$. Hence, we can call RPA recursively with the $n - |G_m|$ remaining agents and fill the remaining $m - 1$ rooms. \square

Table 3 shows an example with $n = 4$ agents and $m = 2$ rectangular rooms. S is the family of rectangles. The total value of both rooms is 5. In each cell, V is the value of that agent to that room and P is the corresponding partner-number. The RPA sends agents 1 and 2 to the right room and agents 3 and 4 to the left room. All agents value their room as at least 2, so after dividing each room between its two agents, the value of each agent is at least 1.¹²

4.3. Procedures for four and three walls

Our procedures for cakes with 4 and 3 walls are recursive and have the following general structure:

- Normalize the total cake value for every agent to $En - F$, where E and F are procedure-specific constants.
- Partition the cake to smaller sub-cakes and issue an **Eval query** on each sub-cake.
- For each agent and sub-cake, calculate the agent's partner-number for the sub-cake. For each agent, the sum of the partner numbers must be at least n . This implies some constraints on the coefficients E and F .
- Run the Room Partition Algorithm (RPA) and get a partition of the agents to groups, one group for each sub-cake. Now there are two cases:
 - *Easy case*: Two or more groups are non-empty. Divide each sub-cake recursively among the agents in the group.
 - *Hard case*: A single group contains all n agents. This means that all agents want a single sub-cake and nobody wants the remainder. Here we cannot divide the desired sub-cake recursively to the n agents because this might lead to endless recursion. Therefore, we must persuade one of

¹²Since the total value of the cake is 5, the proportionality of this division is at least $1/5$. This example can be generalized to show that: $\text{Prop}(\text{Two disjoint rectangles}, \text{Rectangles}, n = 4) \geq 1/5$. Moreover, it is easy to construct an impossibility result, similar to the ones in Section 3, which proves the opposite inequality: $\text{Prop}(\text{Two disjoint rectangles}, \text{Rectangles}, n = 4) \leq 1/5$. This result can be easily generalized to any number of disjoint rectangles and agents.

Room / agent	$1, \dots, n$
top left	$V = 1 - \epsilon \rightarrow P = 0$
top right	$V = 1 - \epsilon \rightarrow P = 0$
bottom left	$V = 4 - \epsilon \rightarrow P = 1$
bottom right	$V = 6(n - 1) - 8 + 3\epsilon \rightarrow P = n - 1$
total	$\sum V = 6n - 8 ; \sum P = n$

Table 4: Data for Four-Quarters-Procedure example: dividing a square cake among n agents.

the agents to take the remainder, so that the desired sub-cake can be divided recursively among $n - 1$ agents. One way to do this is to shrink the desired sub-cake and enlarge the remainder, until one of the agents agrees to take the remainder.

We do this by issuing a **Mark query**. Each agent is asked to make a mark inside the desired sub-cake, such that if the cake is cut by that mark, he is indifferent between sharing the shrunk sub-cake with $n - 1$ agents and taking the remainder alone. The smallest remainder is selected. The agent that made that mark receives the remainder. The other $n - 1$ agents receive the shrunk sub-cake and divide it among them recursively.

With each recursion step, the number of agents in each sub-group is smaller than n . Hence, the procedure eventually terminates with a single agent in each group.

Each agent receives a square with a value of at least 1. Hence the proportionality is at least $1/(En - F)$.

We give here a short description of each procedure. Full details and flow-charts for each procedure are given in the appendices.

4.3.1. Four Quarters Procedure

For presentation purposes, we initially describe a simple procedure with sub-optimal proportionality guarantee. The gory details are given in Appendix A. The reader might find it easier to understand the procedure through the flow-chart in Figure A.11. The procedure is recursive and has two steps:

- (1) **Eval query**: Partition C into 4 quarters in a 2×2 grid and ask each agent to evaluate each quarter. Calculate the partner-numbers for each agent. Run the RPA. If two or more groups are non-empty, divide the sub-cakes recursively to their groups. If a single group contains all n partners, move to the next query:
- (2) **Mark query**: Ask each agent to mark a corner square inside the desired quarter, such that the remainder is an L-shape with a value of exactly 3. Give the smallest L-shape to its drawer. This L-shape can be covered by 3 squares, so by the Covering Lemma its owner can get a square with a value of at least 1. Divide the remaining corner square recursively among the remaining $n - 1$ agents.

As it turns out, the smallest constants that satisfy the constraints implied by the RPA are $E = 6$ and $F = 8$. Hence, for every $n \geq 2$:

$$\text{Prop}(\text{Square}, \text{Squares}, n) \geq \frac{1}{6n - 8}$$

Example 4.2. An example is shown in Table 4. There are $n \geq 3$ agents and they all give the same answers to the Eval query. They evaluate two quarters as $1 - \epsilon$ (where $\epsilon > 0$ is a small constant). This means that their partner numbers for these quarters are 0. The procedure cannot allocate these quarters to any agent and must therefore discard them. The agents evaluate the third quarter as $4 - \epsilon$. This value is too low for two agents: as we already know from our impossibility results, the

value of a square should be at least 4 in order to guarantee a value of 1 to each of two agents. Hence the partner numbers are 1, and the procedure must give this entire quarter to a single agent (the agent that receives this quarter is selected arbitrarily by the RPA, which breaks ties arbitrarily). The remaining $n - 1$ agents value the remaining quarter as $(6n - 8) - (1 + 1 + 4 - 3\epsilon)$. Fortunately, this equals $(6(n - 1) - 8) + 3\epsilon$, which is slightly more than the value required to divide the remaining quarter recursively to the remaining $n - 1$ agents.

The above example shows why E (the coefficient of n in the proportionality expression) must be at least 6. In this example, 6 units of value are wasted while only a single agent receives a piece. Indeed, this example is the worst case of the four quarters procedure; it can be verified that in all other cases, the value loss per agent is smaller. Hence $E = 6$. To calculate F , recall that for $n = 2$ agents, the value should be 4. Hence $E \cdot 2 - F = 4$ so $F = 8$.

We call the coefficient of n the *proportionality coefficient*. The four quarters procedure gives a proportionality coefficient of 6, but the coefficient in the impossibility result is 2 (Section 3). With some more effort, the proportionality coefficient can be improved to 4.

4.3.2. Bounded-Unbounded Procedures

We can decrease the proportionality coefficient to 4 by using a pair of procedures that call each other recursively: a “4 walls procedure”, for dividing a 2-fat rectangle with all sides bounded, and a “3 walls procedure”, for dividing a rectangle with one of its longer sides unbounded (the meaning of an unbounded side is that some of the pieces may flow over the border, even though there is no value over there). Full details and flow charts are given in Appendix B. We obtain the following pair of results $\forall n \geq 2$:

$$\begin{aligned} \text{Prop}(2 \text{ fat rectangle with all sides bounded, squares, } n) &\geq \frac{1}{4n - 4} \\ \text{Prop}(\text{Rectangle with a long side unbounded, squares, } n) &\geq \frac{1}{4n - 5} \end{aligned}$$

Since a square is a 2-fat rectangle, we can fill some of the cells in the table of Sub. 1.3.1:

$$\begin{aligned} \text{Prop}(\text{Square with 4 walls, squares, } n) &\geq \frac{1}{4n - 4} \\ \text{Prop}(\text{Square with 3 walls, squares, } n) &\geq \frac{1}{4n - 5} \end{aligned}$$

Division procedures pieces of other shapes can be found in Appendices D- D.6.

4.3.3. Fat-Thin Procedures: Identical Value Measures

When all agents have the *same* value measure, we can decrease the proportionality coefficient to 2, thus closing the gap with the impossibility result. This again requires a pair of procedures calling each other recursively: a “fat procedure” for dividing a 2-fat rectangle with all sides bounded and a “thin procedure” for dividing a 2-thin rectangle (a rectangle with a length/width ratio of *at least* 2) having one side unbounded. A third “3-walls procedure” can be used for dividing an arbitrary rectangle having one of its long sides unbounded. The details and flow charts are given in Appendix C. We obtain the following pair of results $\forall n \geq 2$:

$$\begin{aligned} \text{PropSame}(2 \text{ fat rectangle with all sides bounded, squares, } n) &\geq \frac{1}{2n} \\ \text{PropSame}(2 \text{ fat rectangle with a long side unbounded, squares, } n) &\geq \frac{1}{2n - 1} \end{aligned}$$

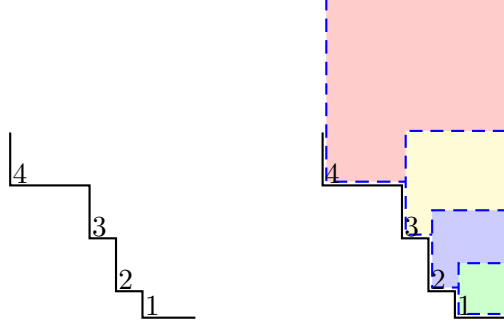


Figure 8: A staircase with $k = 4$ corners (numbered 1,...,4) and its 4 covering quarter-planes, represented by squares in their bottom-left corner.

With which we can fill two more cells in the table of Sub. 1.3.1:

$$\text{PropSame}(\text{Square with 4 walls, squares, } n) \geq \frac{1}{2n}$$

$$\text{PropSame}(\text{Square with 3 walls, squares, } n) \geq \frac{1}{2n-1}$$

Although the situation of n agents with identical value measures seems unrealistic, it may be a good approximation to reality in cases when the cake to be divided is land and every land-piece has a market price, since people usually tend to value land-pieces in correlation with their market price. This motivates the following corollary:

Corollary. Let C be a cake and $\{V_i\}_{i=1}^n$ value measures of n agents on C . For every piece $X \subseteq C$, define:

$$\text{Max}(X) = \max_{i=1,\dots,n} \{V_i(X)\} \quad \text{Min}(X) = \min_{i=1,\dots,n} \{V_i(X)\}$$

$$\text{Ratio}(X) = \frac{\text{Max}(X)}{\text{Min}(X)} \quad r = \max_{X \subseteq C} \text{Ratio}(X)$$

Then there exists a division of C among the n agents such that every agent receives a square with a value of at least:

$$\frac{1}{2nr}$$

Proof. Define a continuous value measure V by: $V(X) = \text{Max}(X) + \text{Min}(X)$. By the procedures of Appendix C, there exists an allocation of n squares in C such that for every square X_i : $V(X_i) \geq \frac{V(C)}{2n}$. Give square X_i to agent i .

Now, by our definitions: $V_i(X_i) \geq \text{Min}(X_i) = \frac{V(X_i)}{\text{Ratio}(X_i)+1} \geq \frac{V(X_i)}{r+1}$ and $V_i(C) \leq \text{Max}(X_i) = V(X_i) - \text{Min}(X_i) \leq \frac{r \cdot V(C)}{r+1}$. Hence, the proportion of agent i is at least $\frac{V_i(X_i)}{V_i(C)} \geq \frac{V(X_i)}{r \cdot V(C)} \geq \frac{1}{2nr}$. \square

This bound is better than that of Subsection 4.3.2 when $r < 2$, i.e. when the maximum value of every land-piece is less than twice the minimum value of the same piece.

4.4. Procedures for two, one and zero walls

When the cake has two walls or less, better proportionality can be guaranteed using a different geometric technique.

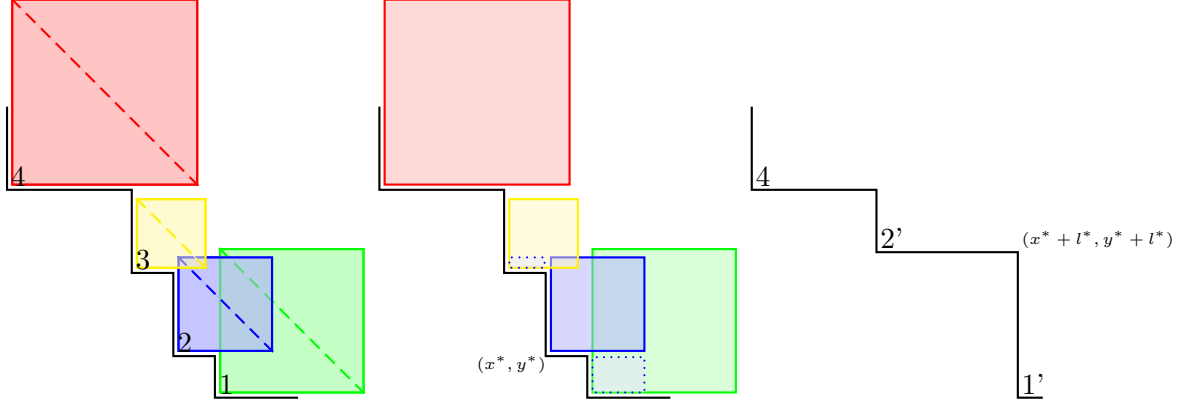


Figure 9: **a.** Corner squares s_i^j (solid) with a diagonal (dashed) representing t_i^j (the taxicab distance from the origin). The square at corner 2 is the winning square as its taxicab distance is minimal (the diagonal is closest to the origin). **b.** The shadow of the winning square (dotted). Note that each rectangular component of the shadow is entirely contained in the square of the corresponding corner. **c.** The staircase after the winning square and its shadow have been removed. Note that corners 1, 2 and 3 are gone and there are two new corners.

We first present a procedure for dividing the top-right quarter-plane, i.e. the cake is a square with two walls and two unbounded sides. Recall from Claim 3.3 that the upper bound for this case is $\frac{1}{2n-1}$. Hence we normalize the value of the entire quarter-plane to $2n - 1$. We begin by giving someone a square at the bottom-left corner, which is worth 1 to him. Ideally, we would like to give the square to an agent who would take the smallest such square, and recurse. However, when we try to do this we run into trouble, as the cake is no longer a quarter-plane.

To run recursively, the procedure must accept a more general input: a *rectilinear polygonal domain with two unbounded sides*, which for brevity we call “staircase” because of its shape (see Figure 8). A staircase can be characterized by its number of inner corners, which we denote by k .

Claim 4.1. If C is a staircase-shaped polygonal domain having k corners, then for every $n \geq 1$:

$$\text{Prop}(C, \text{Squares}, n) = \frac{1}{2n - 2 + k}$$

Proof. We assume that the total value of the staircase C is $2n - 2 + k$. The following procedure can be used to give each agent a square with a value of 1.

(0) If $n = 1$ then let the single agent take the most valuable square in C . Note that the value of C to that agent is at least k and its cover number is k , so by the Covering Lemma it contains a square with a utility of at least 1.

(1) **Mark query:** each agent i marks, in every corner $j \in \{1, \dots, k\}$, a square s_i^j with a value of 1. If no such square exists then the agent draws an infinite square. Note that for every agent the value of the entire cake is at least k , so there must be at least one corner in which the value is at least 1.

(2) For every j , let (x^j, y^j) be the coordinates of corner j . Let l_i^j be the side-length of square s_i^j . For every square s_i^j , calculate $t_i^j := x^j + y^j + l_i^j$.¹³ The *winning square* s^* is a square s_i^j for which t_i^j is smallest (breaking ties arbitrarily). Allocate the winning square to its owner (e.g. in

¹³ t_i^j can be interpreted as the ℓ_1 distance from the origin to the center of s_i^j , or equivalently the ℓ_1 distance to its bottom-right or top-left corner.

Fig. 9(a), the winning square is the square at corner 2).

(3) Remove from C the *shadow* of the winning square by cutting from its top-left corner towards the left boundary of C and from its bottom-right corner towards the bottom boundary of C (e.g. in Fig. 9(b), the two dotted rectangles are removed). Therefore, the total region removed from C , including the winning square and its shadow, is the intersection of C with the rectangle $[0, x^* + l^*] \times [0, y^* + l^*]$.

(4) After the shadow is removed, the remainder is a staircase C' with possibly a different number, k' , of corners (e.g. in Fig. 9(c), $k' = 3$). There are $n' = n - 1$ remaining agents. We prove immediately that all these agents value C' as at least $2n' - 2 + k'$. Divide C' recursively among them. ■

From now on we focus on a single arbitrary agent, with value measure V . After the winning square and its shadow are removed, the cake value changes by ΔV (a negative number), the number of corners changes by Δk and the number of agents changes by $\Delta n = -1$. The claim that we have to prove is: $\Delta V \geq 2\Delta n + \Delta k$, which simplifies to $\Delta V \geq \Delta k - 2$.

We first analyze Δk . When the winning square and its shadow are removed from C , this removes from the cake a certain number $m \geq 1$ of corners - all corners having $(x \leq x^* + l^*)$ and $(y \leq y^* + l^*)$ (e.g. in Fig. 9, $m = 3$ corners are removed - corners 1 2 and 3). Additionally, two new corners are added - one with $(x \leq x^*)$ and $(y = y^* + l^*)$ and one with $(y \leq y^*)$ and $(x = x^* + l^*)$ (e.g. in Fig. 9, corners 1' and 2' are added). Hence, $\Delta k = 2 - m$, and the inequality we have to prove becomes: $\Delta V \geq -m$, or equivalently $|\Delta V| \leq m$.

We now analyze $|\Delta V|$, which is the value contained in the removed region. Partition the removed region to m disjoint rectangular components, starting from the winning square s^* and going to the top-left and to the bottom-right, such that each component is located in a different corner. The components to the top-left of s^* are of the form: $[x, x^*] \times [y, y^* + l^*]$ and the components to its bottom-right are: $[x, x^* + l^*] \times [y, y^*]$. The winning square s^* itself is a single component (e.g. the blue region in Fig. 9/b has three components, in corners 1 2 and 3). To prove that $|\Delta V| \leq m$, it is sufficient to prove that the value of each component is at most 1 (recall that m is the number of removed corners).

Recall that our agent has drawn in each corner a square $[x, x + l] \times [y, y + l]$ with a value of at most 1. Hence, it is sufficient to prove that the component in each corner is contained in the square drawn at that corner. Here we use the fact that all of our agent's squares lost against the winning square in (x^*, y^*) . Hence, for every such square in corner (x, y) : $x + y + l \geq x^* + y^* + l^*$.

First, consider the component s^* itself. Every losing square in corner (x^*, y^*) must have a side-length of at least l^* so it necessarily contains s^* .

Next, consider a component in a corner (x, y) to the top-left of s^* . This is a rectangle: $[x, x^*] \times [y, y^* + l^*]$, with $x < x^*$ and $y < y^* + l^*$. Combining these inequalities with $x + y + l \geq x^* + y^* + l^*$ gives: $y^* + l^* < y + l$ and $x^* < x + l$. Hence the component is contained in the drawn square $[x, x + l] \times [y, y + l]$. The case of a component to the bottom-right of s^* is analogous. □

Corollary 4.1. *By letting $k = 1$ we get:*

$$\text{Prop}(\text{Quarter plane}, \text{Squares}, n) = \frac{1}{2n - 1}$$

A half-plane can be divided by partitioning it to two quarter-planes, to obtain:

Claim 4.2. For every $n \geq 2$:

$$\text{Prop}(\text{Half plane}, \text{Squares}, n) \geq \frac{1}{2n - 2}$$

Proof. Start with a single *mark query*. Assume the cake is the half-plane $y \geq 0$ and there are n agents who value it as $2n - 2$. Ask each agent to draw a vertical line such that the value to the left of the line is exactly 1. Cut the cake at the left-most line and give the entire quarter-plane to the left of the line to the agent who drew that line. The remaining $n - 1$ agents value the remaining quarter-plane as at least $2n - 3 = 2(n - 1) - 1$. Divide it among them using the staircase procedure of Claim 4.1. \square

Similarly we can divide an unbounded plane by partitioning it to two half-planes:

$$\text{Prop}(\text{Plane}, \text{Squares}, n) \geq \frac{1}{2n - 4}$$

4.5. Procedures for compact cakes of any shape

As explained in Subsection 1.3.2, when the cake can be of an arbitrary shape, $\text{Prop}(C, S, n)$ may be arbitrarily small. Hence it makes sense to assess the fairness of an allocation for a particular agent relative to the total utility that this agent can get in an S -piece when given the entire cake. This intuition is captured by the following definition, which is an analogue Definition 2.1, the only difference being that the normalization factor is the cake utility $V^S(C)$ instead of the cake value $V(C)$:

Definition 4.2. (*Relative proportionality*) For a cake C , a family of usable pieces S and an integer $n \geq 1$:

- (a) The *relative proportionality level* of C , S and n , marked $\text{RelProp}(C, S, n)$, is the largest fraction $f \in [0, 1]$ such that, for every set of n value measures (V_i, \dots, V_n) , there exists an S -allocation (X_1, \dots, X_n) for which $\forall i : V_i(X_i)/V_i^S(C) \geq f$.
- (b) The *same-value relative proportionality level* of C , S and n , marked $\text{RelPropSame}(C, S, n)$, is the largest fraction $f \in [0, 1]$ such that, for every single value measure V , there exists an S -allocation (X_1, \dots, X_n) for which $\forall i : V(X_i)/V^S(C) \geq f$.

Our first result involves *axis-parallel squares* - squares whose sides are all parallel to the axes of a given coordinate system, common in urban planning.

Claim 4.3. For every cake C which is a compact subset of \mathbb{R}^2 :

$$\text{RelProp}(C, \text{Axis parallel squares}, n) \geq \frac{1}{8n - 6}$$

Proof. The following allocation procedure can be used.

- (a) Each agent i draws a “best square” in C - a square q_i that maximizes V_i . The existence of such a square can be proved based on the compactness of the set of squares in C ; this is done in Appendix E. By definition of the utility function V^S , for every i : $V_i(q_i) = V_i^S(C)$.
- (b) Virtually create $N = 4n - 3$ copies of each agent i . Divide q_i among these copies using the division procedure for identical value measures described in Appendix C. The result is a collection Q_i of N pairwise-disjoint squares, each of which is worth for agent i at least $V_i(q_i) \cdot \frac{1}{2N} = V_i^S(C) \cdot \frac{1}{8n-6}$. From now on, the original squares q_i are discarded, and we work only with the smaller squares in the collections Q_i .
- (c) Allocate each agent i a single piece from the collection Q_i such that the n allocated pieces are pairwise disjoint. This is done using the following greedy selection procedure:

1. Select a square q^* which is smallest among all squares in all collections. Suppose $q^* \in Q_i$; then allocate q^* to agent i and remove all other squares of Q_i .

2. For each agent $j \neq i$, remove from Q_j all squares that overlap q^* . Since the squares in Q_j are all pairwise-disjoint and not smaller than q^* , the number of squares removed is at most 4. This is based on the following geometric fact: given a certain axis-parallel square q , there are at most 4 axis-parallel squares which are larger than q , overlap q and do not overlap each other (see Figure 10/a). This is because each square larger than q which overlaps q , must overlap one of its 4 corners, so there can be at most 4 such squares.
3. After the removal, each of the remaining $n - 1$ agents has a collection of at least $4(n - 1) - 3$ squares. If only a single agent remains then his collection contains at least 1 square which can be allocated to him; otherwise, we can use the selection procedure recursively.

Now each agent i holds a square with a value of at least $\frac{V_i^S(C)}{8n-6}$, proving the claim. \square

When the squares can be rotated, apparently at most 8 disjoint squares can overlap a smaller square (see Figure 10/b).¹⁴ Writing 8 instead of 4 (and 7 instead of 3) in the proof of Claim 4.3 gives:

Claim 4.4. For every cake C which is a compact subset of \mathbb{R}^2 :

$$\text{RelProp}(C, \text{Squares}, n) \geq \frac{1}{16n - 14}$$

For completeness, we present the following trivial result regarding identical value measures:

Claim 4.5. For every cake C which is a compact subset of \mathbb{R}^2 :

$$\text{RelPropSame}(C, \text{Squares}, n) = \frac{1}{2n}$$

Proof. Suppose the value measure of all n agents is V . Let q be a best square in C - a square that maximizes V . By definition of the utility function, $V(q) = V^S(C)$. Because q is a square, it is possible to allocate within it n disjoint squares with a value of at least $1/2n$. \square

Similar bounds with different constants can be proved for every family of fat pieces; see Appendix D.7.

4.5.1. Remarks

The geometric facts illustrated in Figure 10 have been used for developing approximation procedures for the problem of finding a maximum non-overlapping set [60]. The approximation factors are not tight. For example, for $n = 2$, in step (b) we create $4n - 3 = 5$ axis-parallel squares for each agent, but it is possible to prove that 3 squares per agent suffice for guaranteeing that a pair of disjoint squares exists. Hence, $\text{RelProp}(C, \text{Axis parallel squares}, n = 2) \geq 1/6$. What is the smallest number of squares required to guarantee the existence of n disjoint squares? This open question is interesting because it affects both the proportionality coefficient in our fair cake-cutting

¹⁴We are grateful to Mark Bennet, Martigan, calculus, Red, Peter Woolfitt and Dejan Govc for their help in calculating this number in <http://math.stackexchange.com/q/1085687/29780>.

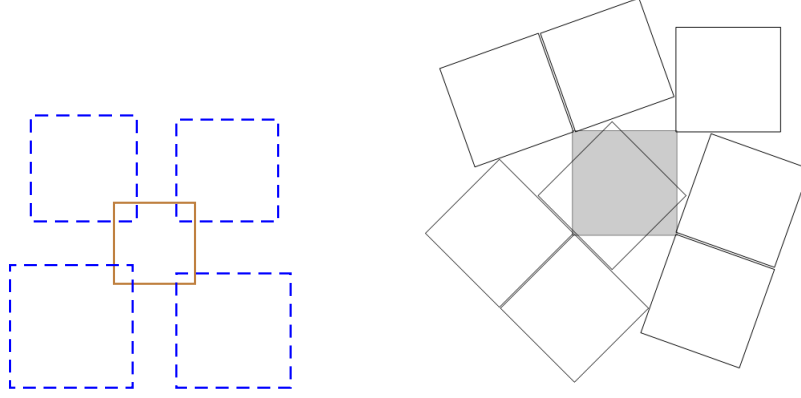


Figure 10: **a.** At most 4 disjoint axis-parallel squares (dashed) can overlap a smaller square (solid). **b.** At most 8 general (rotated) squares can overlap a smaller square. Image credit: Dejan Govc at <http://math.stackexchange.com/a/1393553/29780> . Licensed under CC-BY-SA 3.0 with attribution required.

procedure and the approximation coefficient in the maximum disjoint set algorithm of Marathe et al. [60].

The selection procedure (c) can be used even when the value functions of the agents are not additive or even not monotone (i.e. some parts of the land have negative utility to some agents). As long as every agent can draw N disjoint squares, the procedure guarantees that he receives one of these pieces.

Iyer and Huhns [50] present a division procedure in which each agent marks n desired *rectangles*. Their goal is to allocate each agent a single desired rectangle. However, because the rectangles might be arbitrarily thin, it is possible that a single rectangle will intersect all other rectangles. In this case, the procedure fails and no allocations are returned. In contrast, our procedure requires the agents to draw fat pieces. This guarantees that it always succeeds.

5. Conclusion and Future Work

In this paper we made first steps in an uncharted territory: fair cake-cutting with geometric constraints, inspired by the task of fairly dividing two-dimensional land resources. The new cake-cutting problem introduced here has a large potential for future research. Some possible directions are suggested below.

We would like to close the gaps between the possibility and impossibility results in Tables 1 and 2. Based on our current results, and some other results which we had to omit in order to keep the paper length at a reasonable level, we make the following conjecture:

Conjecture. *When a cake C is divided to n agents each of whom must receive a fat rectangle, the attainable proportionality is:*

$$\frac{1}{2n + \text{Geom}(C)}$$

Where $\text{Geom}(C)$ is a (positive or negative) constant that depends only on the geometric shape of the cake.

In other words: the move from a one-dimensional division to a two-dimensional division asymptotically decreases the fraction that can be guaranteed to every agent by a factor of 2.

The results may be extended to multi-dimensional cakes. Some preliminary results are given in Appendix D.6.

It may be interesting to study cakes of different topologies, such as cylinders and spheres. We mention, in particular, the following challenging open question: *is it possible to divide Earth (a sphere) in a fair-and-square way?*

The present paper focused on *geometric-shape* constraints (squareness or fatness). One could also consider *size* constraints, e.g. by defining the family S to be the family of all rectangles of length above 10 meters or area above 100 square meters. A problem with these constraints is that they are not scalable. For example, if the cake is 200-by-200 meters and there is either a length-minimum of 10 or an area-minimum of 100, then it is impossible to divide the land to more than 400 agents. Governments often cope with this problem by putting an upper bound on the number of people allowed to settle in a certain location. However, this limitation prevents people from taking advantage of new possibilities that become available as the number of people increases. For example, while in rural areas a land-plot of less than 10-by-10 meters may be considered useless because it cannot be efficiently cultivated, in densely populated cities even a land-plot as small as 2-by-2 meters can be used as a parking lot for rent or as a lemonade selling spot. Limiting the number of agents assures that each agent gets a land-plot that can be cultivated efficiently, but it may prevent more profitable ways of using the land-plots. In contrast, the squareness/fatness constraint is scalable because it does not depend on the absolute size of the land-cake. It is equally meaningful in both densely and sparsely populated areas.

The division problem can be extended by allowing each agent to have a different geometric constraint (a different family S of usable shapes) or even to have utility functions which combine different families of usable shapes (with an agent-specific weight for each family).

This paper focuses on the basic fairness criterion of proportionality. We already started to study the stronger criterion of *envy-freeness* [61], using substantially different techniques. In fact, every problem that has been studied with relation to the classic cake-cutting problem, e.g. social welfare maximization and strategy-proofness, can be studied with the additional geometric constraints.

6. Acknowledgments

This research was funded in part by the following institutions:

- The Doctoral Fellowships of Excellence Program at Bar-Ilan University;
- The Adar Foundation of the Economics Department and the Wolfson Chair at Bar-Ilan University;
- The ISF grants 1083/13 and 1241/12;
- The BSF grant 2012344.

These funds had no influence on the contents of the paper and do not necessarily endorse it.

Many members of the stackexchange.com communities kindly helped in coping with specific issues arising in this paper. In particular, we are grateful to the following members: Abel, Abishanka Saha, Acarchau, AcomPi, Adam P. Goucher, Adriano, aes, Alecos Papadopoulos, Andrs Salamon, Anthony Carapetis, Aravind, Barry Cipra, Barry Johnson, BKay, Boris Bukh, Boris Novikov, Brad-Hards, Brian M. Scott, calculus, Chris Culter, Christian Blatter, Claudio Fiandrino, coffeemath, Cowboy, D.W., dafinguzman, Dale M, David Eppstein, David Richerby, Debashish, Dejan Govc, Dmtr Pvlgyi (domotorp), dtldarek, Emanuele Paolini, Flambino, Foo Barrigno, frafl, George Dac-cache, Hagen von Eitzen, Henno Brandsma, Henrik, Henry, Herbert Voss, Ilya Bogdanov, Ittay

Weiss, J.R., Jan Kyncl, Jared Warner, Jeff Snider, John Gowers, John Hughes, JohnWO, Joonas Ilmavirta, Joseph O'Rourke, Kenny LJ, Kris Williams, Macavity, MappaGnosis, Marc van Leeuwen, Mari-Lou A, Mariusz Nowak, Mark Bennet, Martigan, Martin Van der Linden, Matt, Maulik V, Merk, Marcus Hum (mhum), Michael Albanese, Mikal Mayer, Mike Jury, mjqxxxx, mvcouwen, Martin von Gagern (MvG), Nick Gill, njaja, Normal Human, NovaDenizen, Peter Franek, Peter Michor, Peter Woolfitt, pew, PhoemueX, Phoenix, Raphael Reitzig, R Salimi, Raymond Manzoni, Realz Slaw, Red, Robert Israel, Ross Millikan, sds, Stephen, Steven Landsburg, Steven Taschuk, StoneyB, tomasz, TonyK, Travis, vkurchatkin, Wodzimierz Holsztyski, Yury, Yuval Filmus and Zur Luria.

A. Four Quarters Procedure (see Figure A.11)

INPUT: A square cake C and $n \geq 2$ agents. For every agent i : $V_i(C) \geq 6n - 8$.

OUTPUT: For every agent, a square in C whose value is at least 1.

PROCEDURE:

(1) **Eval query:** Partition C into 4 quarters in a 2×2 grid and ask each agent to evaluate the quarters C_j ($j = 1, \dots, 4$). Discretize each agent's values in the following way: (a) If $V_i(C_j) < 1$ then $P_{i,j} = 0$; (b) If $V_i(C_j) > V_i(C) - 3$ then $P_{i,j} = n$; (c) Otherwise $P_{i,j} = \lfloor \frac{V_i(C_j) + 8}{6} \rfloor$. The properties of the floor operator guarantee that $\sum_{j=1}^4 P_{i,j} \geq n$ so the Room Partition Algorithm (RPA) can be used to partition the agents to groups G_j ($j = 1, \dots, 4$). There are two cases:

- *Easy case:* The number of agents in each group is less than n . Then for every agent i in room C_j , one of the following holds: (1) $|G_j| = 1$. By the RPA, this implies $P_{i,j} \geq 1$. By the selection of $P_{i,j}$, this implies $V_i(C_j) \geq 1$. Give C_j to agent i who is the single member of G_j . (2) $2 \leq |G_j| \leq n - 1$. By RPA and the selection of $P_{i,j}$, this implies $V_i(C_j) \geq 6|G_j| - 8$. Divide C_j recursively among the agents in group G_j .
- *Hard case:* All n agents are in the same group, corresponding to a single quarter C_j . By the RPA, this implies that for all agents i : $P_{i,j} \geq n$. By the selection of $P_{i,j}$, this implies $V_i(C_j) > V_i(C) - 3$. Note that here we cannot just divide C_j to the agents in G_j as this might lead to an unbounded recursion. However, we can use the fact that $V_i(C_j) > V_i(C) - 3$ to issue another query:

(2) **Mark query:** each agent i marks an L-shape with a value of exactly 3, whose complement is a corner square with a value of exactly $V_i(C) - 3$ inside the quarter C_j . The procedure gives the smallest L-shape, L_{min} , to its drawer. By the Covering Lemma this L-shape contains a square with a value of at least 1. The remainder, $C \setminus L_{min}$, is corner-square whose value for the remaining $n - 1$ agents is at least $(6n - 8) - 3$. If $n - 1 = 1$ then this value is at least 1 so $C \setminus L_{min}$ can be given to the single remaining agent; otherwise the value is more than $6(n - 1) - 8$, so the square $C \setminus L_{min}$ can be divided recursively among the remaining $n - 1$ agents.

B. Bounded-Unbounded Procedures

We describe a pair of procedures calling each other recursively. Both procedures get as input:

- A cake C which is assumed to be the rectangle $[0, L] \times [0, 1]$;
- n agents, each with a continuous value measure $\{V_i\}_{i=1}^n$ on C .

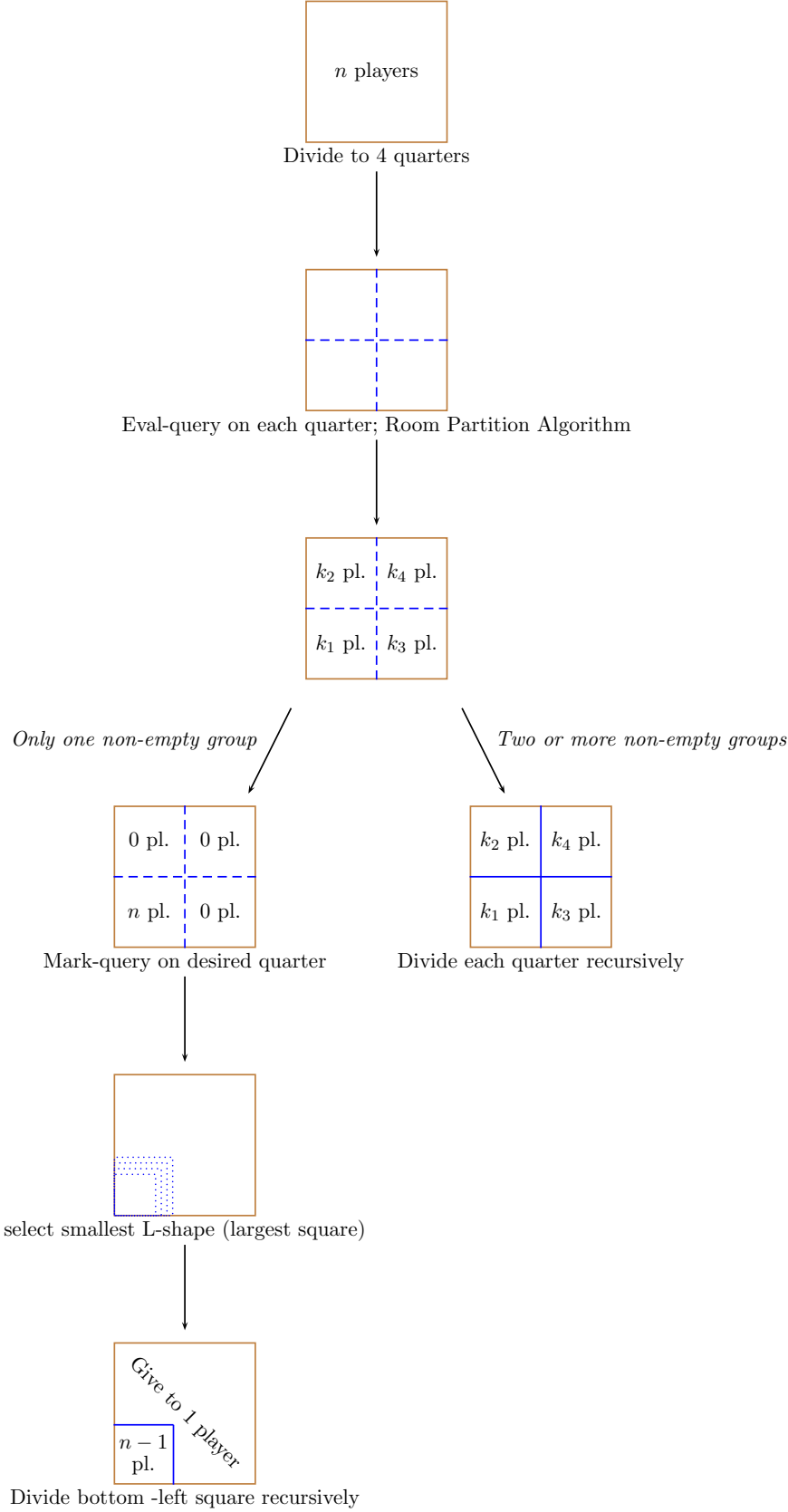


Figure A.11: **Four-quarters procedure.** See Appendix A.

INPUT: A square and n agents that value it as at least $6n - 8$.

OUTPUT: Each agent can get a square with value ≥ 1 .

The numbers k_1, k_2 etc. are the numbers of agents in the groups G_1, G_2 etc. returned by the Room Partition

All procedures return as output n disjoint square pieces $\{X_i\}_{i=1}^n$ such that $\forall i : V_i(X_i) \geq 1$.

The procedures differ in the number of “walls” (bounded sides): in the 4 walls procedure all output squares should be contained in C , while in the 3 walls procedure some output squares may flow over the right border $x = L$, although there is no value in the region $x > L$ (here x refers to the horizontal axis).

B.1. 4 Walls procedure (see Figure B.12)

PRE-CONDITIONS: C is a 2-fat rectangle ($1 \leq L \leq 2$). $\forall i : V_i(C) \geq \max(2, 4n - 4)$. $n \geq 1$.

POST-CONDITION: All n output squares are contained in $C = [0, L] \times [0, 1]$.

PROCEDURE: If $n = 1$, give the entire cake to the single agent. Note that $V(C) \geq 2$ and C is 2-fat, so by the Covering Lemma its utility is at least 1. Otherwise use the following queries:

(1) **Eval query:** Partition C along its longer side to a left half ($x \leq \frac{L}{2}$) and right half. Ask each agent to evaluate each half. Discretize each agent’s values in the following way: (a) If $V_i(C_j) < 2$ then $P_{i,j} = 0$; (b) If $V_i(C_j) > V_i(C) - 2$ then $P_{i,j} = n$; (c) Otherwise $P_{i,j} = \lfloor \frac{V_i(C_j) + 4}{4} \rfloor$. The properties of the floor operator guarantee that $P_{i,1} + P_{i,2} \geq n$ for all i . Hence we can use the Room Partition Algorithm and partition the agents to two groups, G_1 and G_2 , such that for every agent i in group j : $P_{i,j} \geq |G_j|$. There are two cases:

- *Easy case:* The number of agents in each group is less than n . Make a vertical cut at $x = \frac{L}{2}$. The two halves are 2-fat rectangles. By the selection of the $P_{i,j}$ ’s, for every agent i in group j : $V_i(C_j) \geq \max(2, 4|G_j| - 4)$. Using the 4 walls procedure, divide each half to the agents in its group.
- *Hard case:* all n agents are in a single group; w.l.o.g, assume it is the group corresponding to the left half $[0, \frac{L}{2}] \times [0, 1]$. This means that all agents value the right half as at most 2 and the left half as at least $V_i(C) - 2 \geq 4n - 6$. Proceed to the next query:

(2) **Mark query:** Each agent i draws a rectangle adjacent to the right side of the cake: $[x_i, L] \times [0, 1]$, with a value of exactly 2. Let $x^* = \max x_i$. Again there are two cases:

- *Easy case:* $x^* \geq \frac{1}{2}$. Make a vertical cut at x^* . Both parts are 2-fat rectangles. Give the right part to the agent we drew it, since for this agent the right part has a value of 2. Using the 4-walls procedure, divide the left part among the other $n - 1$ agents, since for these agents the left part has a value of at least $4n - 6 > \max(2, 4(n - 1) - 4)$.
- *Hard case:* $\forall i : x_i < \frac{1}{2}$. This means that all agents value the “far left ” of C (the rectangle $x \leq \frac{1}{2}$) as at least $4n - 6$. Proceed to the next query:

(3) **Eval query:** partition the “far left ” of C , the rectangle $[0, \frac{1}{2}] \times [0, 1]$, to two squares: $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ and $[0, \frac{1}{2}] \times [\frac{1}{2}, 1]$, and ask each agent to evaluate each half. Discretize each agent’s values in the following way: (a) If $V_i(C_j) < 1$ then $P_{i,j} = 0$; (b) If $V_i(C_j) > V(C) - 3$ then $P_{i,j} = n$; (c) Otherwise $P_{i,j} = \lfloor \frac{V_i(C_j) + 5}{4} \rfloor$. The properties of the floor operator guarantee that $P_{i,1} + P_{i,2} \geq n$. Thus we can use the Room Partition Algorithm and partition the agents to two groups, G_1 and G_2 , such that for every agent i in group j : $P_{i,j} \geq |G_j|$. There are two cases:

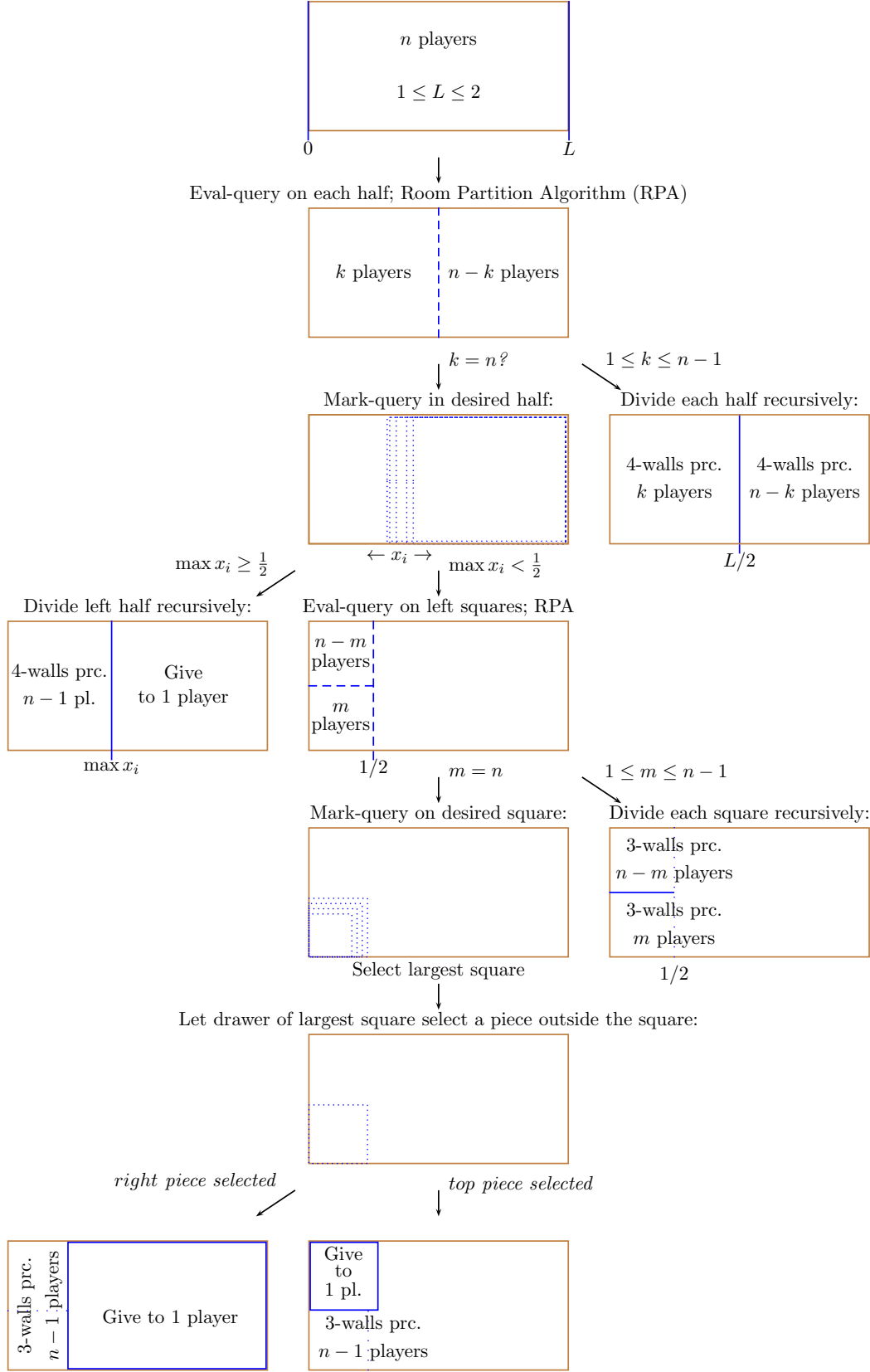


Figure B.12: **4 Walls procedure**. See Appendix B.

INPUT: A 2-fat rectangle; n agents that value it as at least $4n - 4$.

OUTPUT: Each agent receives a square within the 4 walls of the rectangle, with value ≥ 1 .

The 3-walls procedure is illustrated in Figure B.13.

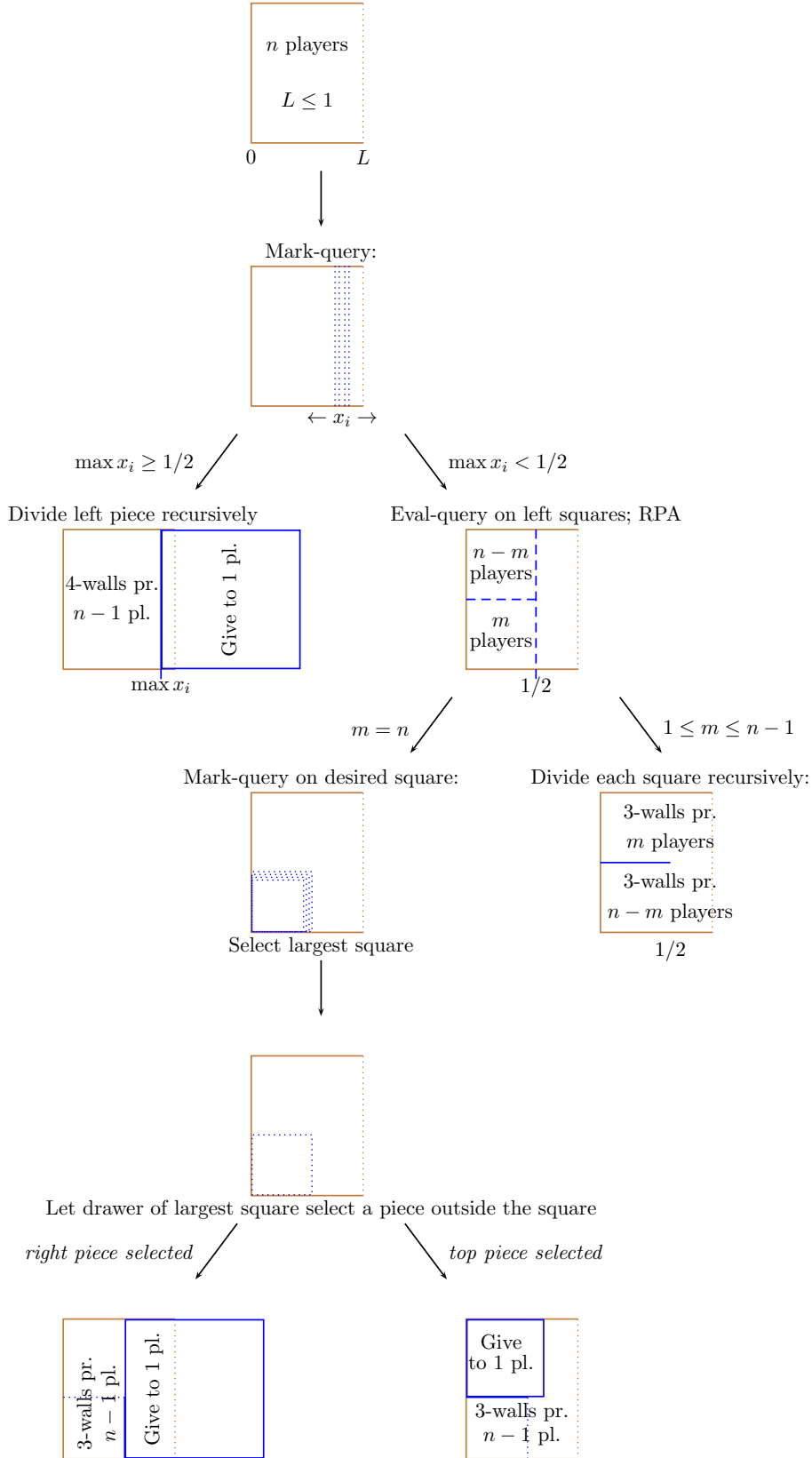


Figure B.13: **3 Walls procedure**. See Appendix B. 30

INPUT: A rectangle; n agents that value it as at least $4n - 5$.

OUTPUT: Each agent receives a square with value ≥ 1 . The squares must be within 3 walls of the rectangle, but may flow over the right side (dotted).

The 4 walls procedure is illustrated in Figure B.12

- *Easy case*: The number of agents in each group is less than n . make a horizontal cut at $y = \frac{1}{2}$. The two halves are squares. By the selection of the $P_{i,j}$'s, for every agent i in group j : $V_i(C_j) \geq \max(1, 4|G_j| - 5)$. Using the *3 walls procedure*, divide each square to the agents in its group. Note that the 3-walls procedure might allocate pieces that flow over the right boundary of the squares (the line $x = \frac{1}{2}$), but this is OK because no square was allocated in the right part of the cake ($x > \frac{1}{2}$).
- *Hard case*: all n agents are in a single group; w.l.o.g, assume it is the group corresponding to the bottom-left square $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$. This means that all agents value the bottom-left as at least $V_i(C) - 3$. Proceed to the next query:

(4) **Mark query**: Each agent i draws a corner square with a value of exactly $V_i(C) - 3$ adjacent to the bottom-left corner. The procedure picks the largest square C_{max} and lets its drawer (hence *the winner*) select a part of its complement $C \setminus C_{max}$. Note that the winner values $C \setminus C_{max}$ as exactly 3. Also note that $C \setminus C_{max}$ can be covered by two pieces: a *top square* (the maximal square between the top of C_{max} and the top of C) and an *right rectangle* (the maximal 2-fat rectangle between the right of C_{max} and the right of C). Thus, at least one of the following must hold:

- The winner values the top square as at least 1; if this is the case then the winner accepts the top square, and the right rectangle remains unallocated.
- The winner values the right rectangle as at least 2; if this is the case then the winner accepts the right rectangle (which is 2-fat), and the top square remains unallocated. If this is the case, then rotate C clockwise 90° , such that the cake to the right of C_{max} is unallocated.

The remaining $n-1$ agents value C_{max} as at least $(4n-4)-3$, which is more than $\max(1, 4(n-1)-5)$. The cake to the right of C_{max} is unallocated, so we can use the *3 walls procedure* to divide C_{max} among them.

B.2. 3 Walls procedure (see Figure B.13)

PRE-CONDITIONS: C is an arbitrary rectangle with $L \leq 1$. For every agent i : $V_i(C) \geq \max(1, 4n-5)$. $n \geq 1$.

POST-CONDITION: All n output squares are contained in $[0, L+1] \times [0, 1]$, i.e.: they are within three walls of C but may flow over the right wall $x = L$.

PROCEDURE: If $n = 1$, give the square $[0, 1] \times [0, 1]$ to the single agent. Note that $V(C) \geq 1$ and C is contained in the allocated square. Otherwise use the following queries:

(1) **Mark query**: Each agent i draws a rectangle adjacent to the right side of the cake: $[x_i, L] \times [0, 1]$, with a value of exactly 1. Let $x^* = \max x_i$. There are two cases:

- *Easy case*: $x^* \geq \frac{1}{2}$. Make a vertical cut at x^* . Give to its drawer the square $[x^*, x^*+1] \times [0, 1]$. Note that this square contains $[x^*, L] \times [0, 1]$ so its value for its drawer is at least 1. The cake to the left of x^* is a 2-fat rectangle and its value of the remaining $n-1$ agents is at least $V(C) - 1 \geq 4n-6 \geq \max(2, 4(n-1)-4)$. Use the *4 walls procedure* to divide it among them.
- *Hard case*: $\forall i : x_i < \frac{1}{2}$. This means that all agents value the “far left” of C (the rectangle $x \leq \frac{1}{2}$) as at least $V(C) - 1 \geq 4n-6$. Proceed to the next query:

(2) **Eval query:** partition the “far left ” of C , the rectangle $[0, \frac{1}{2}] \times [0, 1]$, to two squares: $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ and $[0, \frac{1}{2}] \times [\frac{1}{2}, 1]$, and ask each agent to evaluate each half. Discretize each agent’s values in the following way: (a) If $V(C_j) < 1$ then $P_j = 0$; (b) If $V(C_j) > V(C) - 2$ then $P_j = n$; (c) Otherwise $P_j = \lfloor \frac{V(C_j)+5}{4} \rfloor$. The properties of the floor operator guarantee that $P_1 + P_2 \geq n$. Thus we can use the Room Partition Algorithm and partition the agents to two groups, G_1 and G_2 , such that for every agent in group j : $P_j \geq |G_j|$. There are two cases:

- *Easy case:* The number of agents in each group is less than n . make a horizontal cut at $y = \frac{1}{2}$. The two halves are squares. By the selection of the P_j ’s, for every agent in group j : $V(C_j) \geq \max(1, 4|G_j| - 5)$. Using the *3 walls procedure*, divide each square to the agents in its group.
- *Hard case:* all n agents are in a single group; w.l.o.g, assume it is the group corresponding to the bottom-left square $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$. This means that all agents value the bottom -left as at least $V(C) - 2$. Proceed to the next query:

(3) **Mark query:** Each agent i draws a corner square with a value of exactly $V_i(C) - 2$ adjacent to the bottom-left corner. The procedure picks the largest square C_{max} and lets its drawer (hence *the winner*) select a part of its complement $C \setminus C_{max}$. Note that the winner values $C \setminus C_{max}$ as exactly 2. Also note that $C \setminus C_{max}$ can be covered by two squares: a top *square* (the maximal square between the top of C_{max} and the top of C) and an right *square* (a maximal square to the right of C_{max}). At least one of these squares must have a value of at least 1 to the winner. If the top square has value 1 then give it to the winner and leave the cake to the right of C_{max} unallocated; otherwise, the right square has value 1 : give it to the winner and rotate C clockwise 90° so again the cake to the right of C_{max} is unallocated. The remaining $n - 1$ agents value C_{max} as at least $(4n - 5) - 2$, which is more than $\max(1, 4(n - 1) - 5)$. use the *3 walls procedure* to divide C_{max} among them.

C. Fat-Thin Procedures

In this appendix we prove that for every $n \geq 2$: $\text{PropSame}(2 \text{ fat rectangle}, \text{Squares}, n) \geq 1/2n$. The proof uses a pair of procedures calling each other recursively. The input to both procedures is:

- A cake C which is assumed to be the rectangle $[0, 1] \times [0, L]$ where $L \geq 1$.
- A single continuous value measure V .¹⁵

Both procedures output a certain number of squares such that for each square s : $V(s) \geq 1$.

The procedures differ in the fatness of the input cake (whether it is 2-fat or 2-thin) and in the number of walls (bounded sides): in the fat procedure, all output squares should be contained in C , while in the thin procedure, some output squares may flow over the right border $x = 1$, although there is no value in the range $x > 1$.

C.1. Fat procedure

PRE-CONDITIONS: C is a 2-fat rectangle ($1 \leq L \leq 2$); $V(C) \geq 2n$; $n \geq 1$.

POST-CONDITION: n output squares contained in $C = [0, 1] \times [0, L]$.

PROCEDURE (see Figure C.14):

If $n = 1$, because C can be covered by 2 squares and its value is 2, it necessarily contains a square with a value of 1. ■

Else ($n \geq 2$), for every $u \in [0, 2n]$, define y_u as the largest value $y \in [0, L]$ such that the cake to the bottom of y has value u : $V([0, 1] \times [0, y_u]) = u$. Since $y_{2n} = L \geq 1$, there exists a smallest $k \in [1, n]$ such that: $y_{2k} \geq \frac{1}{2}$. Mark the cake to the bottom of y_{2k} ($[0, 1] \times [0, y_{2k}]$) as *Bottom* and the other possibly empty part ($[0, 1] \times [y_{2k}, L]$) as *Top*. We have $V(\text{Bottom}) = 2k$ and $V(\text{Top}) = 2(n - k)$. Now there are two cases:

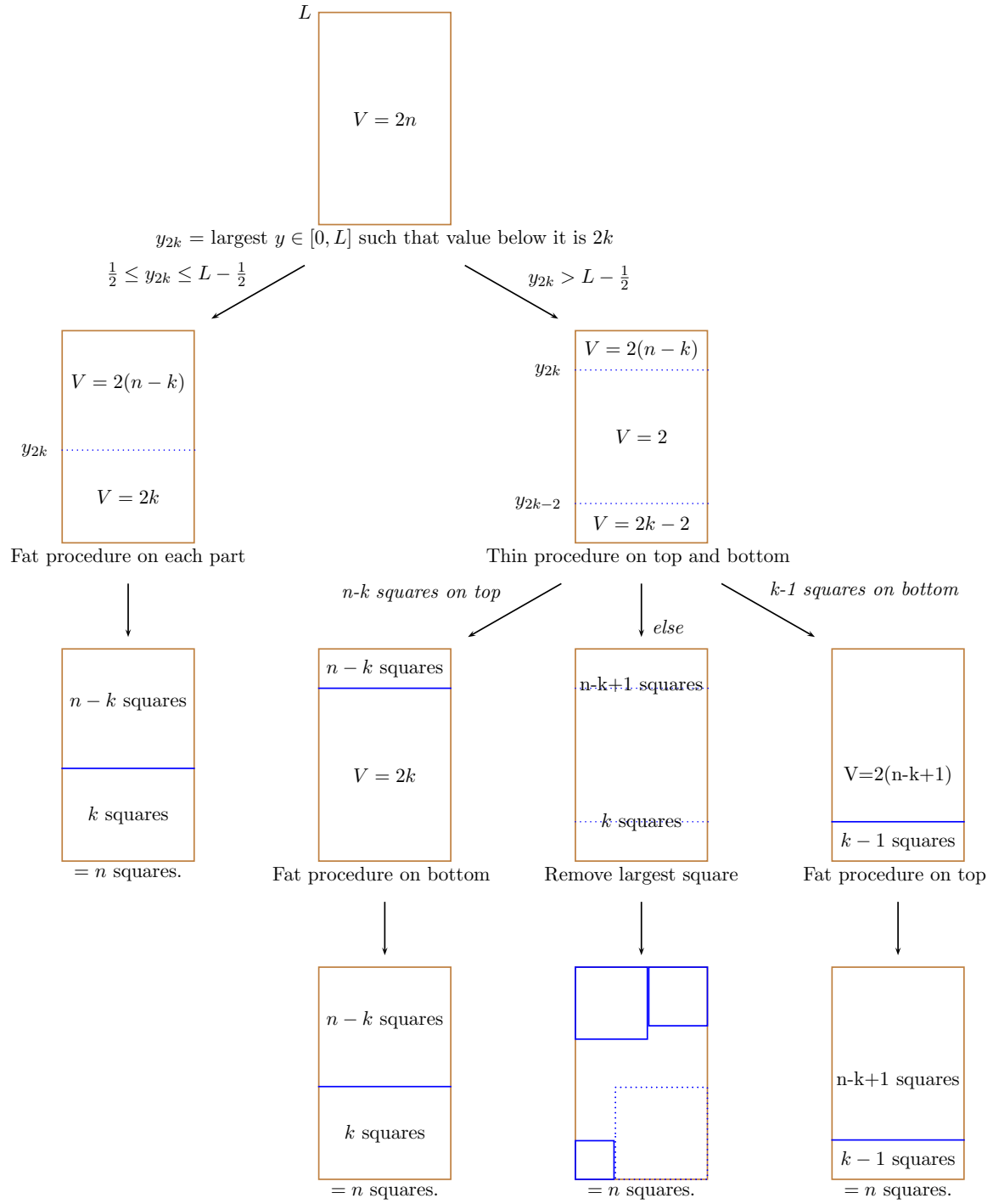
Case A: $L - y_{2k} \geq \frac{1}{2}$ (this implies $k < n$). Thus *Bottom* and *Top* are 2-fat. Recursively run the fat procedure on *Bottom* and *Top* and get $k + (n - k) = n$ squares within the 4 walls of C . ■

Case B: $L - y_{2k} < \frac{1}{2}$, so *Bottom* is 2-fat and *Top* is 2-thin. Now consider y_{2k-2} . By definition of k , $y_{2k-2} < \frac{1}{2}$. let $\text{Bottom}' = [0, 1] \times [0, y_{2k-2}]$ and $\text{Top}' = [0, 1] \times [y_{2k-2}, L]$, so $V(\text{Bottom}') = 2(k - 1) = V(\text{Bottom}) - 2$ and $V(\text{Top}') = 2(n - k + 1) = V(\text{Top}) + 2$. Note that *Bottom'* is 2-thin and *Top'* is 2-fat.

Because here $n \geq 2$, either $n - k \geq 1$ or $k - 1 \geq 1$ or both. Hence, at least one of the two 2-thin pieces (*Top*, *Bottom'*) is non-empty and with value at least 2. Run the *thin procedure* (sec. C.2) on the non-empty thin piece/s and proceed according to the outcome/s:

- If *Top* is non-empty and the procedure on *Top* returns $n - k \geq 1$ squares within the 4 walls of *Top*, then run the fat procedure on *Bottom* getting the remaining k squares. ■
- If *Bottom'* is non-empty and the procedure on *Bottom'* returns $k - 1 \geq 1$ squares within the 4 walls of *Bottom'*, then run the fat procedure on *Top'* getting the remaining $n - k + 1$ squares. ■
- If *Top* is empty ($k = n$) and the procedure on *Bottom'* returns $k = n$ squares within 3 walls of *Bottom'* and contained in $[0, 1] \times [0, 1]$, then we are done. ■

¹⁵We assume that every piece with positive area has positive value. This assumption comes only to simplify the presentation and reduce the number of cases to consider. It can be dropped by adding details to the procedure.



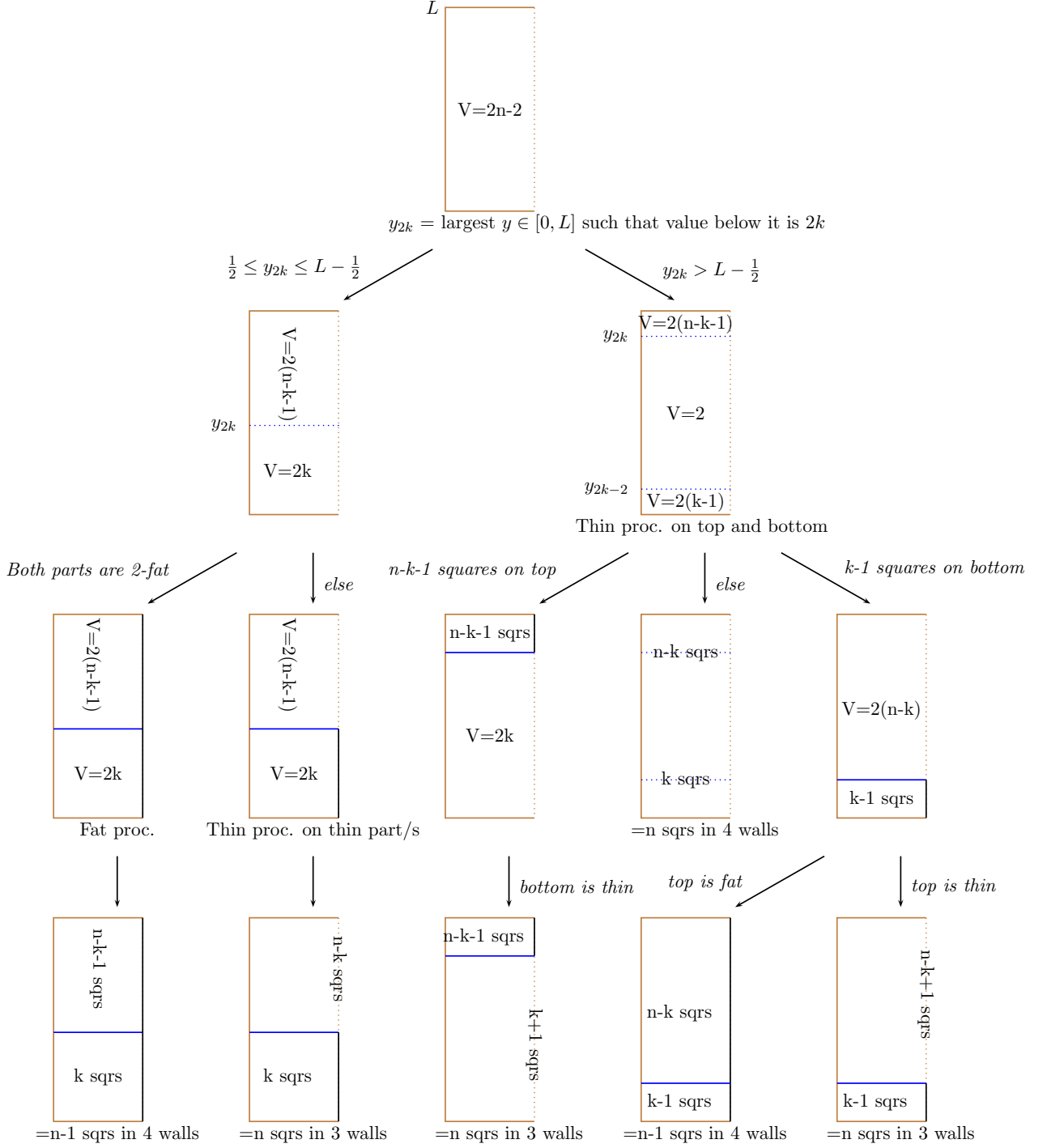


Figure C.15: **Thin procedure.** See Appendix C.

INPUT: A 2-thin rectangle (a rectangle with length/width ratio at least 2), with a value of $2n - 2$.

OUTPUT: $n - 1$ disjoint value-1 squares within the 4 borders,

OR n disjoint value-1 squares within 3 borders (may overflow one of the *longer* borders).

NOTE: The flow-chart shows only representative sub-cases. There are many other sub-cases but they are similar to the ones shown.

- If $Bottom'$ is empty ($k = 1$) and the procedure on Top returns $n - k + 1 = n$ squares within 3 walls of Top and contained in $[0, 1] \times [L - 1, L]$, then we are done. ■

The remaining case is that both Top and $Bottom'$ are non-empty and the thin procedure on both of them returns the larger number of squares. Now we have a total of $n + 1$ squares. Remove the *largest* square and return the remaining n squares. It remains to prove that the n remaining squares do not overlap; we do this now.

C.1.1. Doves and Hawks

We start with $n + 1$ squares in C :

- $k \geq 1$ bottom squares, all disjoint, in $[0, 1] \times [0, 1 - y_{2k-2}]$ (with side-length at most $1 - y_{2k-2}$).
- $n - k + 1 \geq 1$ top squares, all disjoint, in $[0, 1] \times [L - 1 + (L - y_{2k}), L]$ (with side-length at most $1 - (L - y_{2k})$).

In each side there are two types of squares, which we call doves and hawks:

- *Doves* are squares generated by the first outcome of the thin procedure or by recursive calls to the fat procedure. In the bottom, they are all contained in $[0, 1] \times [0, y_{2k-2}]$; in the top, they are contained in $[0, 1] \times [y_{2k}, L]$.
- *Hawks* are squares generated by the second outcome of the thin procedure. In the bottom, they all have their bottom side at $y = 0$ but their top side might have $y \in [y_{2k-2}, 1 - y_{2k-2}]$; in the top, their top side is at $y = L$ but their bottom side might have $y \in [L - 1 + (L - y_{2k}), y_{2k}]$.

We say that a square s *attacks* a square s' if s is larger than s' and s overlaps s' . This is possible only if s and s' are in two opposite sides, since the squares in each side are disjoint.

The doves obviously do not attack each other because $y_{2k-2} < y_{2k}$, so the only possible attacks are: hawks from the top attacking hawks/doves from the bottom, or hawks from the bottom attacking hawks/doves from the top. We now study the properties of the hawks.

Claim C.1. The sum of the side-lengths of all hawks in each side is at most 1.

Proof. The hawks in the bottom are all bounded in $x \in [0, 1]$ and their bottom side is at $y = 0$. Since they do not overlap, the sum of their side-lengths must be at most 1. A similar argument holds for the top hawks. □

An immediate corollary is that at most one hawk from each side has side-length more than $\frac{1}{2}$. We call each of these two hawks (if it exists) the *dangerous hawk*.

After removing the largest square, at most one dangerous hawk remains; It is only this hawk that might attack other squares in the opposite side. We now prove that even this dangerous hawk does not attack other squares.

Claim C.2. No remaining hawk attacks any dove.

Proof. If a hawk from the top attacks a dove from the bottom, then its side-length must be more than $L - y_{2k-2}$. Similarly, if a hawk from the bottom attacks a dove from the top, then its side-length must be more than y_{2k} . In both cases, only for the dangerous hawk might do that. Now there are two cases:

Case 1: $y_{2k} \geq L - y_{2k-2}$. Then also $y_{2k} \geq 1 - y_{2k-2}$. Since all bottom squares have side-length at most $1 - y_{2k-2}$, no bottom hawk can attack a top dove. If the top dangerous hawk attacks a

bottom dove, then its side-length must be more than $L - y_{2k-2}$ so it must be the largest square and it is removed.

Case 2: $y_{2k} < L - y_{2k-2}$. Then also $1 - (L - y_{2k}) < 1 - y_{2k-2} \leq L - y_{2k-2}$. Since all top squares have side-length at most $1 - (L - y_{2k})$, no top hawk can attack a bottom dove. If the bottom dangerous hawk attacks a top dove, then its side-length must be more than y_{2k} so it must be the largest square and it is removed. \square

Claim C.3. No remaining hawk attacks any hawk.

Proof. There are two cases:

Case 1: There is only one hawk (either bottom *or* top) with side-length more than $\frac{1}{2}$. This is the largest square so it is removed. The remaining n squares have side-length at most $\frac{1}{2}$ and thus do not attack each other.

Case 2: There are two hawks (bottom *and* top) with side-length more than $\frac{1}{2}$. W.l.o.g, assume the top hawk is the largest, with side-length is $d_{north} \geq d_{south}$. By Claim C.1, the sum of the side-lengths of all other top hawks is at most $1 - d_{north}$, hence the side-length of any single other top hawk is at most $1 - d_{north} \leq 1 - d_{south} \leq L - d_{south}$. Hence, the bottom side of all remaining top hawks is at $y \geq d_{south}$. Hence the remaining bottom hawk cannot attack any of them. \square

C.2. Thin procedure

PRE-CONDITIONS: C is a 2-thin rectangle ($2 < L$); $V(C) \geq 2n - 2$; $n \geq 2$.

POST-CONDITION: One of the following two outcomes:

- $n - 1$ output squares within the 4 walls of the input rectangle $C = [0, 1] \times [0, L]$, or -
- n squares within 3 walls of the input rectangle, contained in $[0, L - 1] \times [0, L]$ with their left side at $x = 0$.

PROCEDURE (see Figure C.15):

If $n = 2$ then select $y \in [0, L]$ such that $V([0, 1] \times [0, y]) = V([0, 1] \times [y, L]) = 1$. Proceed according to the value of y :

- If $y \in [1, L - 1]$ then return the two squares $[0, y] \times [0, y]$ and $[0, L - y] \times [y, L]$. Both squares are in $[0, L - 1] \times [0, L]$ with their left side at $x = 0$; this is an instance of the second outcome. \blacksquare
- If $y \in [0, 1)$ then return $[0, 1] \times [0, 1]$; if $y \in (L - 1, L]$ then return $[0, 1] \times [L - 1, L]$. Both cases are instances of the first outcome as the returned square is within the 4 walls of C . \blacksquare

Else ($n \geq 3$), for every $u \in [0, 2n - 2]$, define y_u as the largest value $y \in [0, L]$ such that the cake to the bottom of y has value u : $V([0, 1] \times [0, y_u]) = u$. By our assumptions, $y_0 = 0$ and $y_{2n-2} = L$, so there exists a smallest $k \in [1, n - 1]$ such that: $y_{2k} \geq \frac{1}{2}$. Mark the cake to the bottom of y_{2k} ($[0, 1] \times [0, y_{2k}]$) as *Bottom* and the other part ($[0, 1] \times [y_{2k}, L]$) as *Top*. We have $V(\text{Bottom}) = 2k$ and $V(\text{Top}) = 2(n - k - 1)$. Now there are two cases:

Case A: $L - y_{2k} \geq \frac{1}{2}$ (this implies $k < n - 1$). Thus each of *Bottom* and *Top* is either 2-fat, or 2-thin with the longer side facing right. Recursively run the fat procedure or the thin procedure, whichever is appropriate, on *Bottom* and *Top*. The output is either: $k + (n - k - 1)$ squares within 4 walls, or $k + (n - k)$ or $(k + 1) + (n - k - 1)$ squares within 3 walls. Because both *Bottom* and *Top* are shorter than C , if the thin procedure run on them returns the second outcome, the returned squares will all be within $[0, L - 1] \times [0, L]$ with their left side at $x = 0$. \blacksquare

Case B: $L - y_{2k} < \frac{1}{2}$, so *Bottom* is 2-fat or 2-thin facing right, and *Top* is 2-thin facing bottom. Now consider y_{2k-2} . By definition of k , $y_{2k-2} < \frac{1}{2}$. let $\text{Bottom}' = [0, 1] \times [0, y_{2k-2}]$ and

$Top' = [0, 1] \times [y_{2k-2}, L]$, so $V(Bottom') = 2(k - 1) = V(Bottom) - 2$ and $V(Top') = 2(n - k) = V(Top) + 2$. Note that $Bottom'$ is 2-thin facing top, and Top' is 2-fat or 2-thin facing east.

Note that because here $n \geq 3$, either $n - k - 1 \geq 1$ or $k - 1 \geq 1$ or both. Hence, at least one of the two thin pieces facing bottom /top (Top , $Bottom'$) is non-empty and with value at least 2. Run the *thin procedure* (sec. C.2) on the non-empty piece/s facing bottom /top and proceed according to the outcome:

- If the procedure on Top returns $n - k - 1 \geq 1$ squares within the 4 walls of Top , then run the 4-walls procedure or the thin procedure on $Bottom$, getting the remaining k squares within 4 walls or $k + 1$ squares within 3 walls. Because $Bottom$ is shorter than C , the $k + 1$ squares are contained in $[0, L - 1] \times [0, L]$ with their left side at $x = 0$. ■
- If the procedure on $Bottom'$ returns $k - 1 \geq 1$ squares within the 4 walls of $Bottom'$, then run the 4-walls procedure on Top' getting the remaining $n - k$ squares within 4 walls or $n - k + 1$ squares within 3 walls. Because Top is shorter than C , the $k + 1$ squares are contained in $[0, L - 1] \times [0, L]$ with their left side at $x = 0$. ■

Otherwise, we have the following squares:

- $k \geq 1$ bottom squares in $[0, 1] \times [0, 1 - y_{2k-2}]$;
- $n - k \geq 1$ top squares in $[0, 1] \times [L - 1 + (L - y_{2k}), L]$.

Because $L \geq 2$, no squares overlap. We have n squares within the 4 walls of C , which is an instance of the first outcome. ■

C.3. 3-walls procedure

PRE-CONDITIONS: C is an arbitrary rectangle with $L \leq 1$. $V(C) \geq 2n - 1$. $n \geq 1$.

POST-CONDITION: All n output squares are contained in $[0, L + 1] \times [0, 1]$, i.e.: they are within three walls of C but may flow over the right wall $x = L$.

PROCEDURE: If $n = 1$, give the square $[0, 1] \times [0, 1]$ to the single agent. Note that $V(C) \geq 1$ and C is contained in the allocated square. ■

If $n \geq 2$, then let $x \in [0, L]$ be the largest value such that the square to its right has a value of exactly 1: $V([x, 1 + x] \times [0, 1]) = 1$. The remainder $[0, x] \times [0, 1]$ has a value of exactly $2n - 2$. There are two cases:

- If $x \geq 0.5$ then the remainder is a 2-fat rectangle. Run the fat procedure to get $n - 1$ squares to the left of x . Together with the square $[x, 1 + x] \times [0, 1]$, we have n squares with a value of 1 in $[0, L + 1] \times [0, 1]$.
- If $x < 0.5$ then the remainder is a 2-thin rectangle. Run the *thin procedure* (sec. C.2) and proceed according to the outcome:

- If the procedure returns n squares, then finish. ■
- If the procedure returns $n - 1$ squares within the 4 walls of $[0, x] \times [0, 1]$, then add the square $[x, 1 + x] \times [0, 1]$ and finish. ■

$$\text{Prop}(C, 2, R\text{-fat-rects}) \leq 1/3$$

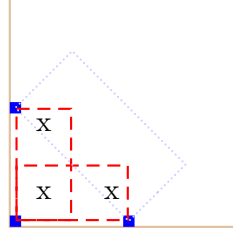


Figure D.16: Impossibility result in a quarter-plane cake with fat rectangle pieces. See Claim D.1.

D. Non-Square Pieces

In the main body of the paper, the family of usable pieces has been the family of squares. This appendix demonstrates the generality of the techniques developed in the paper by presenting results with some alternative families of usable pieces.

D.1. Fat rectangles - impossibility results

Claim 3.1 holds as-is for R -fat rectangles:

Claim D.1. For every $R \geq 1$:

$$\text{PropSame}(\text{Quarter plane}, R \text{ fat rectangles}, 2) \leq 1/3$$

Proof. Let I_3 be the arrangement of 3 pools from the proof of Claim 3.1 (Figure D.16). Every R -fat rectangle touching the two bottom pools must have a height of at least $(10 - 2\epsilon)/R$ and thus, when ϵ is sufficiently small, it must contain the point $(5/R, 5/R)$ and the point $(10 - 10/R, 5/R)$. Every R -fat rectangle touching the two left pools must contain the point $(5/R, 5/R)$ and the point $(5/R, 10 - 10/R)$. And every R -fat rectangle touching the top-left and the bottom-right pools must contain $(10 - 10/R, 5/R)$ and $(5/R, 10 - 10/R)$. See Figure D.16. Hence, in every allocation of disjoint R -fat rectangles, at most one rectangle touches two or more pools. The agent receiving this rectangle has a utility of at most $1/3$. \square

Claim 3.3 is based on Claim 3.1, so it holds as-is for R -fat rectangles. The same is true for the 3-walls result and for the 1-wall claims 3.5 and 3.7. We obtain:

Claim D.2. For every $R \geq 1$:

$$\text{PropSame}(\text{Square with 1 wall}, R \text{ fat rectangles}, n) \leq \frac{1}{1.5n - 2}$$

$$\text{PropSame}(\text{Square with 2 walls}, R \text{ fat rectangles}, n) \leq \frac{1}{2n - 1}$$

$$\text{PropSame}(\text{Square with 3 walls}, R \text{ fat rectangles}, n) \leq \frac{1}{2n - 1}$$

Claim 3.2 does not hold as-is, but the following slightly weaker result follows immediately from Claim D.2 (since adding walls cannot increase the proportionality):

Claim D.3. For every $R \geq 1$:

$$\text{PropSame}(\text{Square with 4 walls}, R \text{ fat rectangles}, n) \leq \frac{1}{2n - 1}$$

By classic cake-cutting protocols, $\text{PropSame}(\text{Square}, \infty \text{ fat rectangles}, n) = 1/n$ (an ∞ -fat rectangle is just an arbitrary rectangle). The PropSame function is thus discontinuous at $R = \infty$. If the agents agree to use any rectangular piece, they can receive their proportional share of $1/n$, but if they insist on using R -fat rectangles, even when R is very large, they might have to settle for about half of this share.

D.2. Fat rectangles - procedures

In Subsection 4.2 we presented a procedure for dividing a square with between two agents giving each agent a square piece. The proportionality guarantee was $1/4$. The following procedure achieves a proportionality guarantee of $1/3$ by allowing the pieces to be 2-fat rectangles. We assume that the total value of the cake is 3. See flow-chart in Figure D.17

a. **Eval query:** partition the cake to two halves and ask each agent to evaluate the halves. For each agent, at least one half must be worth at least 1.

- If one agent values one half as at least 1 and the other agent values the other half as at least 1, then give each half to the agent who values it as at least 1. Done.

- Otherwise, there is only one half (say, the left half) which both agents value as at least 1. This means that they value the right half as less than 1. This means that they value the left half as at least 2. proceed to the next query.

b. **Eval query:** partition the valuable half (e.g. the left half) to two quarters and ask each agent to evaluate the two quarters. For each agent, at least one quarter must be worth at least 1.

- If one agent values one quarter as at least 1 and the other agent values the other quarter as at least 1, then give each quarter to the agent who values it as at least 1. Done.

- Otherwise, there is only one quarter which both agents value as at least 1. Proceed to the next query.

c. **Mark query:** ask each agent to draw inside the valuable quarter and adjacent to the corner of C , a square with value *exactly* 1.

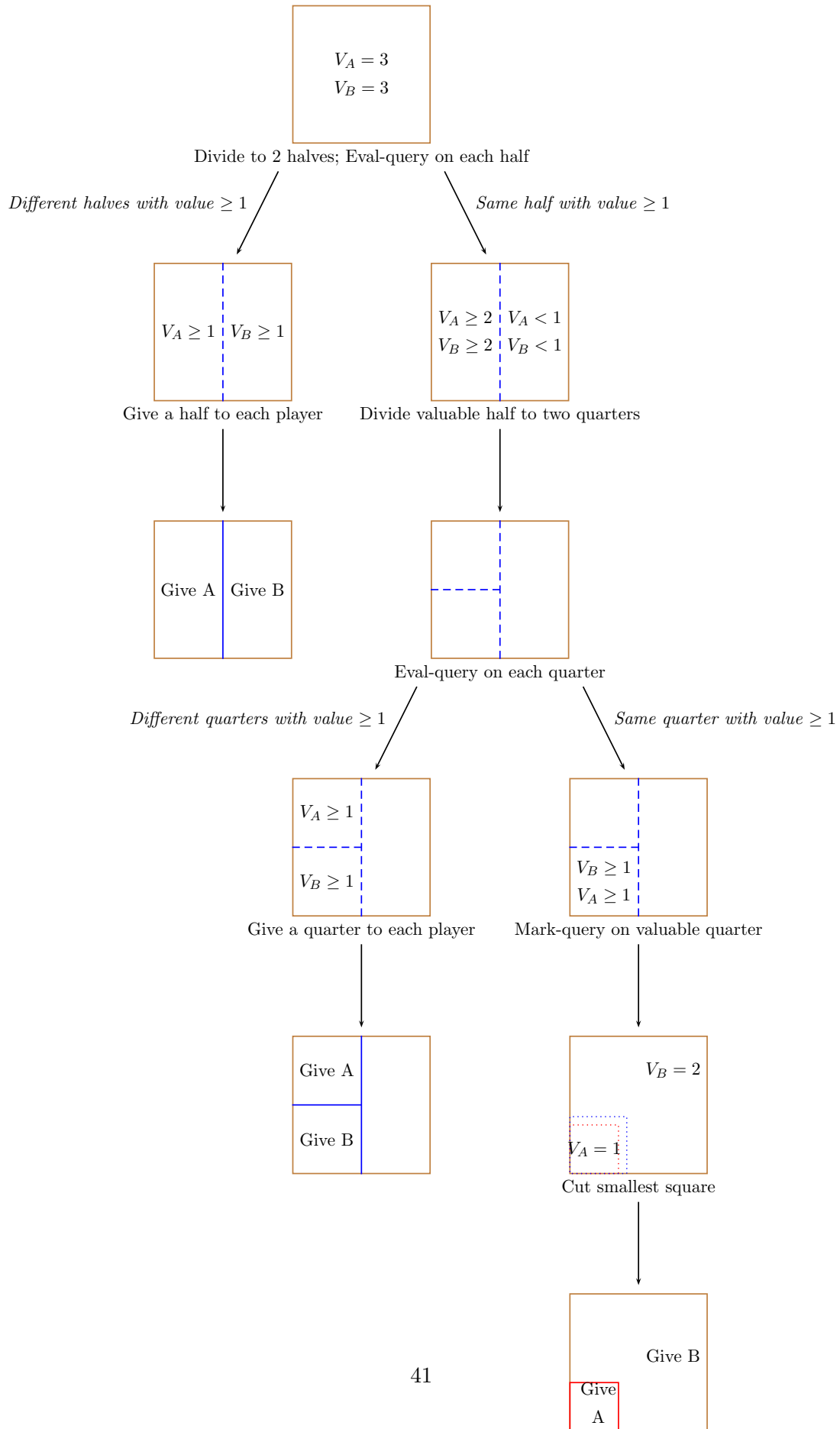
Cut the smaller of the two squares and give it to the agent who drew it (breaking ties arbitrarily). This agent now holds a square which he values as exactly 1. The other agent values the allocated square as at most 1 and the remainder as at least 2. The remainder is an L-shape which can be covered by two 2-fat-rectangles, so by the Covering Lemma it contains a 2-fat-rectangle with a utility of at least 1; give it to the remaining agent and finish.

Combining the lower bound proved by the procedure with the upper bound implied by Claim D.3 gives a tight result for 2 agents:

$$\text{Prop}(\text{Square}, 2 \text{ fat rectangles}, n = 2) = 1/3$$

We can plug this procedure into the Four Quarters Procedure. Consider the worst-case example presented in Sub. 4.3.1. There are two adjacent quarters with a value of less than 1. If the sum of their values is more than 1, we can give them to a single agent since they are a 2-fat rectangle. Hence, the worst case is that the sum of their values is slightly less than 1. Regarding the third quarter, if its value is more than 3 then we can give it to two agents. Hence the worst case is that its value is slightly less than 3 so we must give it to a single agent. All in all, the maximum value loss per agent is $1+3=4$. It can be verified that this case is still the worst case - in all other cases the value loss per agent is at most 4. Hence $E = 4$. By solving $E \cdot 2 - F = 3$, we get that $F = 5$. Hence:

$$\text{Prop}(\text{Square}, 2 \text{ fat rectangles}, n) \geq \frac{1}{4n - 5}$$



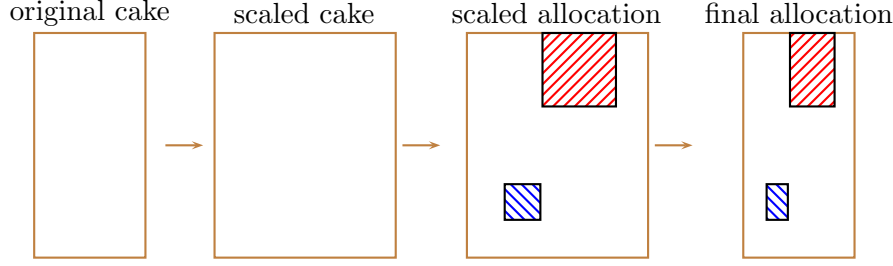


Figure D.18: Giving each agent a fair golden-ratio rectangle.

D.3. Golden-ratio rectangles

In some cases it may be desirable to get rectangles with an exact length/width ratio, rather than just an upper bound as in R -fat rectangles. For example, when dividing advertisement areas in a newspaper or website, for aesthetic reasons it may be desirable to give each agent a piece with a length/width ratio equal to the golden ratio $\Phi = 1.618\dots$

If all agents agree on a single optimal length/width ratio, then the problem can be solved easily by scaling one axis of the coordinate system in that ratio. For example, if the original cake is a 2-by-1 rectangle and all agents want golden-ratio rectangles, then we can define a coordinate system in which the land-cake is a 2-by- Φ rectangle and then divide the cake using a square-based procedure. The square will be translated to a golden-ratio rectangle in the original coordinate system (see Figure D.18).¹⁶

By using R -fat rectangle pieces instead of square pieces, it is possible to get an $x \times y$ rectangle such that $\frac{\Phi}{R} \leq \frac{x}{y} \leq \Phi R$. Thus, it is also possible to require a range of length/width ratios, as long as the *geometric mean* of the range is equal for all agents.

What if different agents want length/width ratio ranges with different geometric means? Is it possible, for example, that some agents get golden-ratio rectangles while other agents get squares with the same partial-proportionality guarantee? This question is still open.

D.4. Pairs of squares

In this subsection we allow to give each agent a union of two squares. As a background, note that in the cake-cutting literature there are two types of divisions:

1. Divisions in which each agent gets a single connected piece;
2. Divisions in which each agent gets an arbitrary number of disjoint pieces.

But the common practice in land division is a compromise between these two extremes. Each agent can receive more than one land-plot, but not an unlimited number of small land-plots - usually 2 or 3 land-plots per agent. As far as we know, this scenario has not been studied yet in the cake-cutting literature.

In the context of geometric constraints, we assumed that divisions are of type #1. Type #2 divisions are uninteresting in the context of geometric constraints, because every geometric shape can be approximated by a sufficiently large number of squares. Therefore, every division of a cake (particularly, a proportional division) can be approximated to an arbitrary precision as a type #2

¹⁶The resulting rectangles are all aligned such that their long side is parallel to the long side of the original land-cake. If we want the opposite, we should scale the other axis of the coordinate system, such that the land-cake is a $2\Phi \times 1$ rectangle.

division where each agent gets a sufficiently large set of disjoint squares. However, giving two plots per agent *can* improve the proportionality guarantee. In the following paragraphs we allow to give each agent a union of two (possibly overlapping) squares.

Consider first the case of two agents. The following procedure, which is a variant of the procedure of Subsection 4.2, achieves a proportionality guarantee of $1/2$. We assume that the total value of the square cake is 2. See flow-chart in Figure D.19

a. **Eval query**: partition the cake to four quarters. Pair them diagonally, i.e, the top-left with the bottom-right and the top-right with the bottom-left. Ask each agent to evaluate the two pairs. For each agent, at least one pair must be worth at least 1. If each agent values a different pair as at least 1, then give each pair to the agent who wants it and finish. Otherwise, go to the next query.

b. **Mark query**: ask each agent to draw, inside the valuable pair and adjacent to the corners of C , a pair of squares with the same side-length and with a value of exactly 1 (this can be seen as “shrinking” the pair of squares by an identical factor).

Cut the smaller of the two pairs and give it to the agent who drew it (breaking ties arbitrarily). This agent now holds a union of two *disjoint* squares which he values as exactly 1. The other agent values the remainder as at least 1. The remainder is a union of two *overlapping* squares; give it to the remaining agent and finish.

Thus, for $n = 2$ people, a proportional division is possible when square-pairs are allowed:

$$\text{Prop}(\text{Square}, \text{Pair of squares}, 2) = 1/2$$

We can plug this procedure into the Four Quarters Procedure. Consider the worst-case example presented in Sub. 4.3.1. There are two quarters with a value of less than 1. If the sum of their values is more than 1, we can give them to a single agent. Hence, the worst case is that the sum of their values is slightly less than 1. Regarding the third quarter, if its value is more than 2 then we can give it to two agents. Hence the worst case is that its value is slightly less than 2. All in all, the maximum value loss per agent is $1+2=3$. Hence $E = 3$. By solving $E \cdot 2 - F = 2$, we get that $F = 4$. Hence:

$$\text{Prop}(\text{Square}, \text{Pair of squares}, n) \geq \frac{1}{3n - 4}$$

This raises some interesting questions, such as: can this bound can be further improved? What is the upper bound on proportionality with pairs of pieces? What happens when every agent can get a union of k squares when $k \geq 3$? We leave these questions to future research.

D.5. 45-degree polygons

In this subsection we expand the family of usable pieces to include, in addition to squares, also other 2-fat polygons with angles that are multiples 45 degrees. We call such polygons *2-FFDPs* (2-fat Forty-Five Degree Polygons).¹⁷ Our procedure is based on the following geometric facts:

1. A right-angled isosceles triangle (RAIT) is a 2-FFDP.
2. Both a RAIT and a square can be partitioned into two congruent halves, each of which is a RAIT.
3. Each RAIT half in such a partition can be shrunk by translating the division line towards one of the corners, such that the smaller piece is a RAIT and the larger piece is a 2-FFDP.

¹⁷This idea was suggested to the first author by Galya Segal-Halevi in private communication.

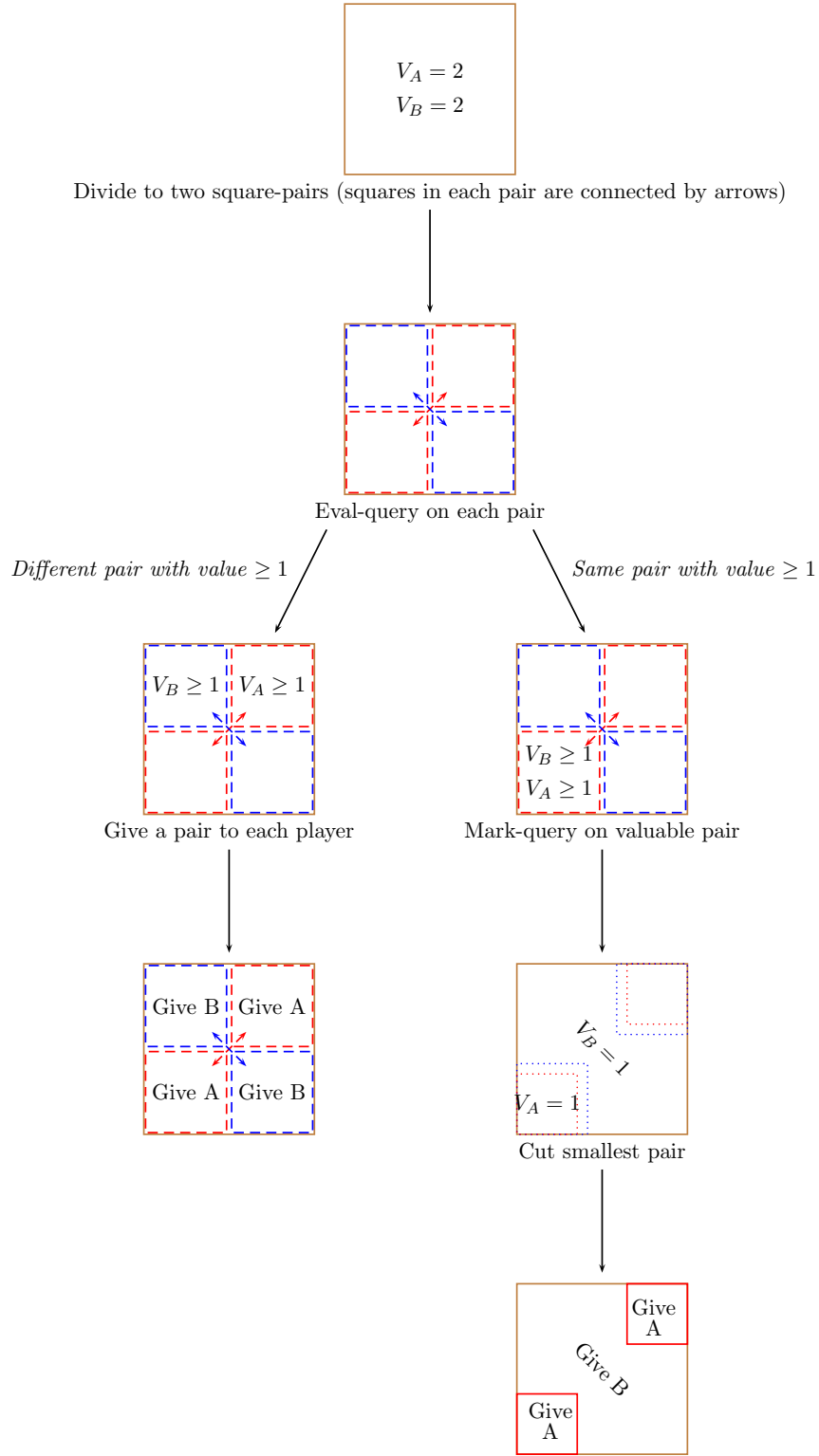


Figure D.19: Dividing a square to two agents who want square-pairs.

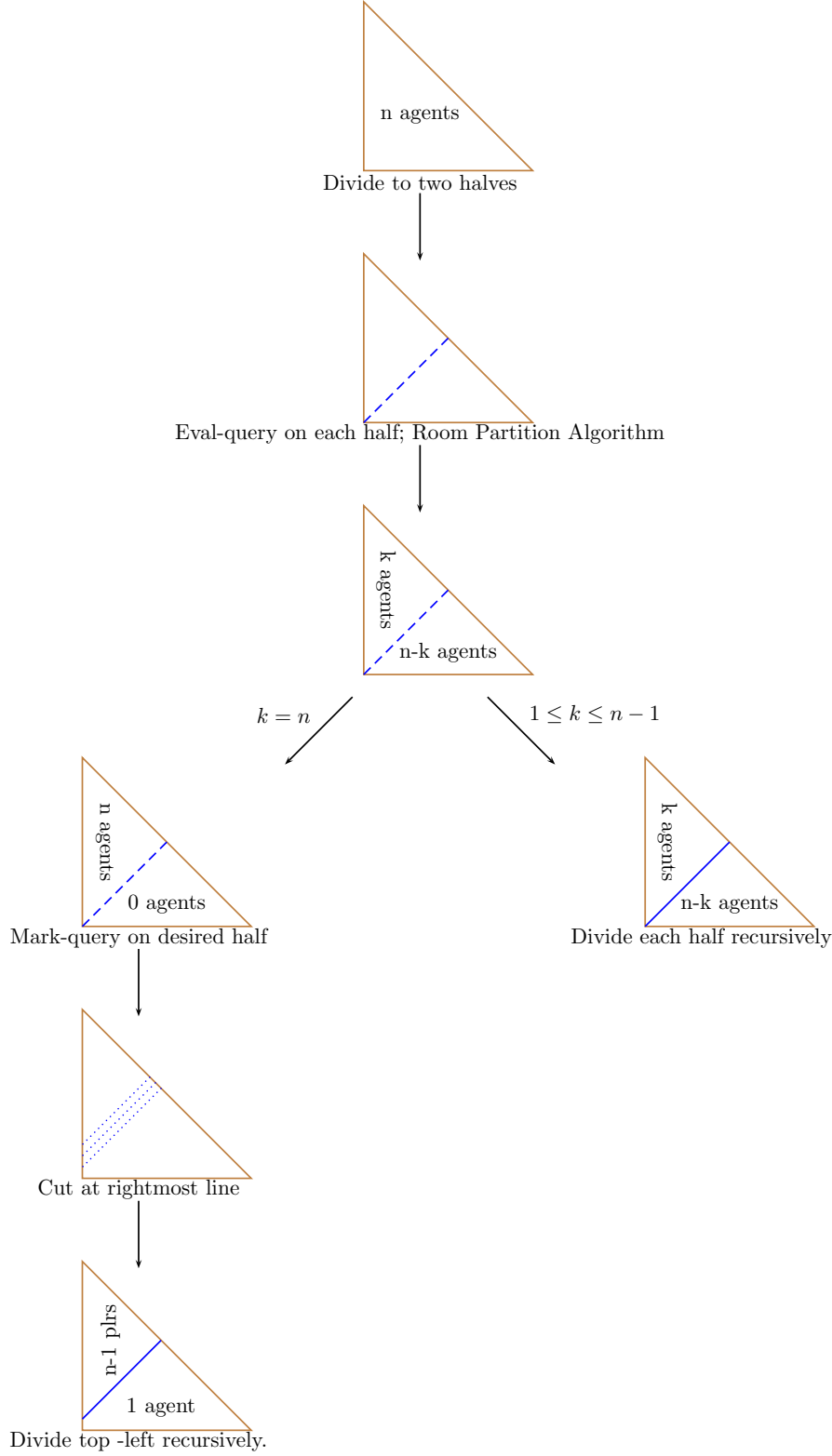


Figure D.20: Dividing a right-angled isosceles triangle (RAIT) to 2-fat 45-degree polygons (2-FFDPS). See Claim D.4.

INPUT: A right-angled isosceles triangle; n agents that value it as at least $2n - 2$.

OUTPUT: Each agent can get a 2-fat 45-degree polygon (a RAIT or a quadrangle) with value ≥ 1 .

Using the eval-mark procedure illustrated Figure D.20 for RAIT cakes, and a similar procedure for square cakes, it is easy to prove the following two results:

Claim D.4. For every $n \geq 2$:

$$\text{Prop}(\text{RAIT}, 2 \text{ FFDPs}, n) \geq \frac{1}{2n-2}$$

$$\text{Prop}(\text{Square}, 2 \text{ FFDPs}, n) \geq \frac{1}{2n-2}$$

The procedure is clearly much simpler than when the pieces must be squares (as in Appendix B) and the proportionality coefficient is better. In other words, it is easier to divide a cake fairly when 45-degree polygons are allowed. This might explain why practical land allocation maps usually contain more than just rectangles.

D.6. Multi-dimensional cubes

In this subsection we briefly mention a generalization of our results to multi-dimensional cubes.

Consider first the generalization of a quarter-plane: a d -dimensional area which is bounded in one direction in each of the d dimensions. By arranging pools as in Claim 3.3, we get d pools for every new agent. Hence:

$$\text{PropSame}(C, d \text{ cubes}, n) \leq \frac{1}{d(n-1)+1}$$

If C is a cube bounded in all d dimensions, then instead of the last d pools we can add $2^d - 1$ pools at the corners. This gives the following generalization of Claim 3.4:

$$\text{PropSame}(C, d \text{ cubes}, n) \leq \frac{1}{d(n-2)+2^d}$$

(in particular, when $n = 2$ there is a single pool in each of the 2^d corners. For each new agent, one of the pools is deflated towards its corner and d new pools are added adjacent to it). These two extreme cases are generalized by:

Claim D.5. Let C be a d -cube with δ unbounded sides, all in different dimensions (i.e. at most one side is unbounded in each dimension). Then $\forall n \geq 2$:

$$\text{PropSame}(C, d \text{ cubes}, n) \leq \frac{1}{d(n-2)+(2^{d-\delta}+\delta)}$$

On the positive side, we can generalize the recursive procedure of Appendix A. We have a different procedure for each number of unbounded sides; the procedures call each other recursively:

Claim D.6. Let C be a d -dimensional cube with $\delta \leq d$ unbounded sides, all in different dimensions (i.e. at most one side is unbounded in each dimension), and $2d - \delta$ walls. Then $\forall n \geq 2$:

$$\text{Prop}(C, d \text{ cubes}, n) \geq \frac{1}{(2^{d-1}+2)(n-2)+(2^{d-\delta}+\delta)}$$

We do not know how to generalize the staircase technique of Claim 4.1 to three or more dimensions. Closing the gap between the upper and lower bounds in multi-dimensional cakes is another one of our open questions.

D.7. Arbitrary Fat Shapes

The proof of Claim 4.3 can be generalized to arbitrary fat shapes, based on the following geometric lemma [62]:

Lemma D.1. *Let S be a family of d -dimensional R -fat pieces (where $d \geq 1$ and $R \geq 1$ are finite constants). Let $s \in S$ and let Q be a set of pairwise-disjoint elements of S , all with diameter at least as large as s . There is a finite constant O , which depends only on S , such that the number of objects in Q overlapping s is at most O . [62]*

We call this constant $\text{Overlap}(S)$. For example, we saw in Figure 10 that $\text{Overlap}(\text{Parallel Squares}) = 4$ and $\text{Overlap}(\text{Squares}) = 7$. Note that the lemma is not true when the objects are not fat. For example, $\text{Overlap}(\text{Rectangles}) = \infty$, because a rectangle having its longer side vertical can intersect an arbitrarily large number of rectangles having their longer side horizontal.

Lemma D.1 and similar results have been used for developing constant-factor approximation procedures for the NP-hard problem of finding a maximum non-overlapping set [60]. We use the same lemma to prove the following general claim:

Claim D.7. Let S be a family of fat pieces, $O = \text{Overlap}(S)$ and $\text{PropSame}(S, S, n) = \inf_{C \in S} \text{PropSame}(C, S, n)$. Then for every closed and bounded cake C and every $n \geq 1$:

$$\text{RelProp}(C, S, n) \geq \text{PropSame}(S, S, On - O + 1)$$

Proof. Copy the proof of Claim 4.3, changing only the following steps:

- (b) Set $N = On - O + 1$. Get a collection Q_i of N pairwise-disjoint squares whose value for agent i is at least $V_i(s_i) \cdot \text{PropSame}(S, S, N)$.
- (c) After allocating a smallest piece, at most O pieces of other agents have to be removed. \square

Claims 4.3 and 4.4 can be attained as special cases of Claim D.7 by setting $O = 4$ or $O = 8$, respectively. By calculating $\text{Overlap}(S)$ for different families of fat pieces, it is possible to prove analogous claims with different constants.

For example, when S is the family of R -fat rectangles, $\text{Overlap}(S) = 2 \lceil R \rceil + 2$. Substituting in Claim D.7 gives:

$$\text{RelProp}(C, R \text{ fat rectangles}, n) \geq \frac{1}{(4 \lceil R \rceil + 4)(n - 1) + 2}$$

Finally, when all agents have the same value measure, we get:

Claim D.8. For every cake C which is a compact subset of \mathbb{R}^2 and for every family S :

$$\text{RelPropSame}(C, S, n) = \text{PropSame}(S, S, n)$$

Proof. Suppose the value measure of all n agents is V . Let q be a best S -piece in C - an S -piece that maximizes V . By definition of the utility function: $V(q) = V^S(C)$. Because $q \in S$, it is possible to allocate n S -pieces in q with a value of at least $V(q) \cdot \text{PropSame}(S, S, n)$. \square

E. Existence of Best pieces

This appendix shows how to prove the existence of a usable piece with a maximum value (this is used in the proof of Claim 4.3). We start by defining a metric space of pieces (recall that a *piece* is a Borel subset of \mathbb{R}^2 and *Area* is its Lebesgue measure).

Definition E.1. The *symmetric difference (SD) pseudo-metric* is defined by:

$$d_{SD}(X, Y) = \text{Area}[(X \setminus Y) \cup (Y \setminus X)]$$

d_{SD} is not a metric because there may be different pieces whose symmetric difference has an area of 0, e.g., a square with an additional point and a square with a missing point. To make SD a metric, we consider only pieces X that are *regularly open*, i.e., the interior of the closure of themselves: $X = \text{Int}[\text{Cl}[X]]$.

Claim E.1. SD is a metric on the set of all regularly-open pieces.

*Proof.*¹⁸ Let X and Y be two regularly-open sets such that $d_{SD}(X, Y) = 0$. We prove that $X = Y$.

$d_{SD}(X, Y) = 0$ implies $\text{Area}[X \setminus Y] = \text{Area}[Y \setminus X] = 0$.

$Y \subseteq \text{Cl}[Y]$ so $X \setminus Y \supseteq X \setminus \text{Cl}[Y]$. Hence also $\text{Area}[X \setminus \text{Cl}[Y]] = 0$.

X is open and $\text{Cl}[Y]$ is closed; hence $X \setminus \text{Cl}[Y]$ is open (it is an intersection of two open sets).

The only open set with an area of 0 is the empty set (because any non-empty open set contains a ball with a positive measure). Hence: $X \setminus \text{Cl}[Y] = \emptyset$.

Equivalently: $X \subseteq \text{Cl}[Y]$.

By taking the Cl of both sides: $\text{Cl}[X] \subseteq \text{Cl}[Y]$

By a symmetric argument: $\text{Cl}[Y] \subseteq \text{Cl}[X]$

Hence: $\text{Cl}[Y] = \text{Cl}[X]$

By taking the Int of both sides and by the fact that they are regularly-open: $Y = X$. \square

Thus when we allocate a square we actually allocate only its interior. This has no effect on the utility of the agents since the boundary has an area of 0 and so its value is 0 for all agents.

Claim E.2. Let D be the metric space defined by d_{SD} . Let V be a measure absolutely continuous with respect to area. Then V is a uniformly continuous function from D to \mathbb{R} .

Proof. The fact that V is an absolutely continuous measure implies that, for every $\epsilon > 0$ there is a $\delta > 0$ such that every piece X with $\text{Area}(X) < \delta$ has $V(X) < \epsilon$ [63, Proposition 15.5 on page 251]. Hence, for every two pieces X and Y , if $d_{SD}(X, Y) < \delta$ then $\text{Area}(X \setminus Y) < \delta$ and $\text{Area}(Y \setminus X) < \delta$, then $V(X \setminus Y) < \epsilon$ and $V(Y \setminus X) < \epsilon$, then $|V(X) - V(Y)| = |V(X \setminus Y) - V(Y \setminus X)| < \epsilon$. \square

Claim E.3. Let V be a measure absolutely continuous with respect to area and Q a set of pieces which is compact in the SD metric space. Then there exists a piece $q \in Q$ for which V is maximized.

Proof. By the previous claim, V is a uniformly continuous and hence a continuous real-valued function. By the extreme value theorem, it attains a maximum in every compact set. \square

¹⁸We are thankful to Tony K., Phoemue X., Dafin Guzman, Henno Brandsma and Ittay Weiss for contributing to this proof via discussions in the math.stackexchange.com website (<http://math.stackexchange.com/a/1099461/29780>).

The value measures considered in this paper are always absolutely continuous with respect to area. Hence, to prove that a certain set of pieces Q contains a “best piece” it is sufficient to prove that Q is compact. We do this now for the special case in which Q is the set of open squares contained in a given cake (note that the same proof could be used for the set of closed squares):

Claim E.4. Let C be a closed, bounded subset of \mathbb{R}^2 . Let Q be the set of all open squares contained in C . Then Q is compact in the SD metric space.

Proof. It is sufficient to prove that Q is sequentially compact, i.e. every infinite sequence of open squares in C has a subsequence converging to an open square in C . Let $\{q_i\}_{i=1}^\infty$ be an infinite sequence of open squares in C . For every q_i , let (A_i, B_i) be a pair of opposite corners. Because C is compact, it contains $Cl[q]$ and hence contains the points A_i and B_i . Hence the infinite sequence of pairs of points, $\{(A_i, B_i)\}_{i=1}^\infty$, is an infinite sequence in $C \times C$. $C \times C$ is compact because it is a finite product of compact sets. Hence, the sequence has a subsequence converging to a limit point $(A^*, B^*) \in C$. From now on we assume that $\{(A_i, B_i)\}_{i=1}^\infty$ is that converging subsequence. Let q^* be the open square having A^* and B^* as two opposite corners. We show that: (a) q^* is an open square in C ; (b) The subsequence $\{q_i\}_{i=1}^\infty$ converges to q^* .

(a) q^* is obviously an open square by definition. We have to show that each point in q^* is also a point of C . To every square q_i , attach a local coordinate system in which corner A_i has coordinates 0, 0 and corner B_i has coordinates 1, 1 and every other point in $Cl[q_i]$ has coordinates in $[0, 1] \times [0, 1]$. For every coordinate $(x, y) \in [0, 1] \times [0, 1]$, let $q_i(x, y)$ be the unique point with these coordinates in $Cl[q_i]$ (e.g. $A_i = q_i(0, 0)$ and $B_i = q_i(1, 1)$). The sequence $\{q_i(x, y)\}_{i=1}^\infty$ is a sequence of points which are all in C , and they converge to $q^*(x, y)$. Since C is closed, $q^*(x, y) \in C$.

(b) For every i , the area of the symmetric difference between q^* and q_i is bounded and satisfies the following inequality:

$$d_{SD}(q^*, q_i) \leq 4 \cdot \max(d(A^*, A_i), d(B^*, B_i)) \cdot \max(d(A^*, B^*), d(A^*, B_i), d(A_i, B^*), d(A_i, B_i))$$

Since all distances are bounded and $d(A^*, A_i)$, $d(B^*, B_i)$ converge to 0, the same is true for $d_{SD}(q^*, q_i)$. Hence, the subsequence $\{q_i\}_{i=1}^\infty$ converges to q^* .

The previous paragraph proved that Q is sequentially compact. Hence it is compact. \square

In a similar way it is possible to prove similar results for other families S , such as the family of R -fat rectangles or cubes.

References

References

- [1] H. Steinhaus, The problem of fair division, *Econometrica* 16 (1) (1948) 101–104, URL <http://www.jstor.org/stable/1914289>.
- [2] D. Weller, Fair division of a measurable space, *Journal of Mathematical Economics* 14 (1) (1985) 5–17, ISSN 03044068, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0304-4068\(85\)90023-0](http://dx.doi.org/10.1016/0304-4068(85)90023-0).
- [3] S. J. Brams, A. D. Taylor, Fair Division: From Cake-Cutting to Dispute Resolution, Cambridge University Press, ISBN 0521556449, URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0521556449>, 1996.

- [4] F. E. Su, Rental Harmony: Sperner’s Lemma in Fair Division, *The American Mathematical Monthly* 106 (10) (1999) 930–942, ISSN 00029890, secondoftwo doi:, URL <http://www.math.hmc.edu/~f{su}/papers.dir/rent.pdf>.
- [5] J. B. Barbanel, S. J. Brams, Cake division with minimal cuts: envy-free procedures for three persons, four persons, and beyond, *Mathematical Social Sciences* 48 (3) (2004) 251–269, ISSN 01654896, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.mathsocsci.2004.03.006>.
- [6] Y. J. Cohler, J. K. Lai, D. C. Parkes, A. D. Procaccia, Optimal Envy-Free Cake Cutting, in: *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)*, 626–631, URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/download/3638/3938>, 2011.
- [7] X. Bei, N. Chen, X. Hua, B. Tao, E. Yang, Optimal Proportional Cake Cutting with Connected Pieces, in: *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12)*, 1263–1269, URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/download/4980/5456>, 2012.
- [8] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, The Efficiency of Fair Division, *Theory of Computing Systems* 50 (4) (2012) 589–610, ISSN 1432-4350, secondoftwo doi:, URL <http://dx.doi.org/10.1007/s00224-011-9359-y>.
- [9] E. Mossel, O. Tamuz, Truthful Fair Division, in: S. Kontogiannis, E. Koutsoupias, P. G. Spirakis (Eds.), *SAGT*, vol. 6386, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-16169-8, 288–299, secondoftwo doi:, URL http://dx.doi.org/10.1007/978-3-642-16170-4_25, 2010.
- [10] Y. Chen, J. K. Lai, D. C. Parkes, A. D. Procaccia, Truth, justice, and cake cutting, *Games and Economic Behavior* 77 (1) (2013) 284–297, ISSN 08998256, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.geb.2012.10.009>.
- [11] R. Cole, V. Gkatzelis, G. Goel, Mechanism Design for Fair Division: Allocating Divisible Items Without Payments, in: *Proceedings of the Fourteenth ACM Conference on Electronic Commerce, EC ’13*, ACM, New York, NY, USA, ISBN 978-1-4503-1962-1, 251–268, secondoftwo doi:, URL <http://arxiv.org/abs/1212.1522>, 2013.
- [12] J. M. Robertson, W. A. Webb, *Cake-Cutting Algorithms: Be Fair if You Can*, A K Peters/CRC Press, first edn., ISBN 1568810768, URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1568810768>, 1998.
- [13] J. B. Barbanel, *The Geometry of Efficient Fair Division*, Cambridge University Press, ISBN 0521842484, URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0521842484>, 2005.
- [14] S. J. Brams, *Mathematics and Democracy: Designing Better Voting and Fair-Division Procedures*, Princeton University Press, first edition edn., ISBN 0691133212, URL <http://www.sciencedirect.com/science/article/pii/S089571770800174X>, 2007.
- [15] A. D. Procaccia, Cake Cutting Algorithms, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia (Eds.), *Handbook of Computational Social Choice*, chap. 13, Cambridge University Press, URL <http://www.cs.cmu.edu/~arielpro/papers/cakechapter.pdf>, 2015.

- [16] M. Berliant, T. Raa, A foundation of location theory: Consumer preferences and demand, *Journal of Economic Theory* 44 (2) (1988) 336–353, ISSN 00220531, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0022-0531\(88\)90008-7](http://dx.doi.org/10.1016/0022-0531(88)90008-7).
- [17] M. Berliant, W. Thomson, K. Dunz, On the fair division of a heterogeneous commodity, *Journal of Mathematical Economics* 21 (3) (1992) 201–216, ISSN 03044068, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0304-4068\(92\)90001-n](http://dx.doi.org/10.1016/0304-4068(92)90001-n).
- [18] J. Legut, J. A. M. Potters, S. H. Tijs, Economies with Land - A Game Theoretical Approach, *Games and Economic Behavior* 6 (3) (1994) 416–430, ISSN 08998256, secondoftwo doi:, URL <http://dx.doi.org/10.1006/game.1994.1024>.
- [19] C. P. Chambers, Allocation rules for land division, *Journal of Economic Theory* 121 (2) (2005) 236–258, ISSN 00220531, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.jet.2004.04.008>.
- [20] M. Dall’Aglio, F. Maccheroni, Disputed lands, *Games and Economic Behavior* 66 (1) (2009) 57–77, ISSN 08998256, secondoftwo doi:, URL [http://eprints.luiss.it/727/1/Dall’Aglio_no.58_2007.pdf](http://eprints.luiss.it/727/1/Dall'Aglio_no.58_2007.pdf).
- [21] F. Hüsseinov, A theory of a heterogeneous divisible commodity exchange economy, *Journal of Mathematical Economics* 47 (1) (2011) 54–59, ISSN 03044068, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.jmateco.2010.12.001>.
- [22] A. Nicolò, Perea, P. Roberti, Equal opportunity equivalence in land division, *SERIEs - Journal of the Spanish Economic Association* 3 (1-2) (2012) 133–142, secondoftwo doi:, URL <http://dx.doi.org/10.1007/s13209-011-0069-0>.
- [23] W. Stromquist, How to Cut a Cake Fairly, *The American Mathematical Monthly* 87 (8) (1980) 640–644, ISSN 00029890, secondoftwo doi:, URL <http://dx.doi.org/10.2307/2320951>.
- [24] A. Nicolò, Y. Yu, Strategic divide and choose, *Games and Economic Behavior* 64 (1) (2008) 268–289, ISSN 08998256, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.geb.2008.01.006>.
- [25] Y. Azrieli, E. Shmaya, Rental harmony with roommates, *Journal of Economic Theory* 153 (2014) 128–137, ISSN 00220531, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.jet.2014.06.006>.
- [26] D. R. Woodall, Dividing a cake fairly, *Journal of Mathematical Analysis and Applications* 78 (1) (1980) 233–247, ISSN 0022247X, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0022-247x\(80\)90225-5](http://dx.doi.org/10.1016/0022-247x(80)90225-5).
- [27] M. Berliant, K. Dunz, A foundation of location theory: existence of equilibrium, the welfare theorems, and core, *Journal of Mathematical Economics* 40 (5) (2004) 593–618, ISSN 03044068, secondoftwo doi:, URL [http://dx.doi.org/10.1016/s0304-4068\(03\)00077-6](http://dx.doi.org/10.1016/s0304-4068(03)00077-6).
- [28] P. K. Agarwal, M. J. Katz, M. Sharir, Computing depth orders for fat objects and related problems, *Computational Geometry* 5 (4) (1995) 187–206, ISSN 09257721, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0925-7721\(95\)00005-8](http://dx.doi.org/10.1016/0925-7721(95)00005-8).

- [29] M. J. Katz, 3-D vertical ray shooting and 2-D point enclosure, range searching, and arc shooting amidst convex fat objects, *Computational Geometry* 8 (6) (1997) 299–316, ISSN 09257721, secondoftwo doi:, URL [http://dx.doi.org/10.1016/s0925-7721\(96\)00027-2](http://dx.doi.org/10.1016/s0925-7721(96)00027-2).
- [30] H. Moulin, Uniform externalities: Two axioms for fair allocation, *Journal of Public Economics* 43 (3) (1990) 305–326, ISSN 00472727, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0047-2727\(90\)90003-z](http://dx.doi.org/10.1016/0047-2727(90)90003-z).
- [31] E. Budish, The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes, *Journal of Political Economy* 119 (6) (2011) 1061–1103, ISSN 00223808, secondoftwo doi:, URL <http://dx.doi.org/10.1086/664613>.
- [32] A. D. Procaccia, J. Wang, Fair Enough: Guaranteeing Approximate Maximin Shares, in: *Proceedings of the Fifteenth ACM Conference on Economics and Computation, EC '14*, ACM, New York, NY, USA, ISBN 978-1-4503-2565-3, 675–692, secondoftwo doi:, URL <http://www.cs.cmu.edu/~arielpro/papers/mms.pdf>, 2014.
- [33] S. Bouveret, M. Lemaître, Characterizing conflicts in fair division of indivisible goods using a scale of criteria, *Autonomous Agents and Multi-Agent Systems* (2015) 1–32secondoftwo doi:, URL <http://dx.doi.org/10.1007/s10458-015-9287-3>.
- [34] S. Even, A. Paz, A note on cake cutting, *Discrete Applied Mathematics* 7 (3) (1984) 285–296, ISSN 0166218X, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0166-218x\(84\)90005-2](http://dx.doi.org/10.1016/0166-218x(84)90005-2).
- [35] G. J. Woeginger, J. Sgall, On the complexity of cake cutting, *Discrete Optimization* 4 (2) (2007) 213–220, ISSN 15725286, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.disopt.2006.07.003>.
- [36] C. R. Plott, A Notion of Equilibrium and Its Possibility Under Majority Rule, *The American Economic Review* 57 (4) (1967) 787–806, URL <http://resolver.caltech.edu/CaltechAUTHORS:20140220-101651369>.
- [37] H. G. Johnson, Trade and growth : A geometrical exposition, *Journal of International Economics* 1 (1) (1971) 83–101, ISSN 00221996, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0022-1996\(71\)90017-1](http://dx.doi.org/10.1016/0022-1996(71)90017-1).
- [38] J. R. Hines, J. C. Hlinko, T. J. F. Lubke, From each according to his surplus: Equiproportionate sharing of commodity tax burdens, *Journal of Public Economics* 58 (3) (1995) 417–428, ISSN 00472727, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0047-2727\(94\)01489-b](http://dx.doi.org/10.1016/0047-2727(94)01489-b).
- [39] E. Cantillon, A. Rangel, A graphical analysis of some basic results in social choice 19 (3) (2002) 587–611, secondoftwo doi:, URL <http://dx.doi.org/10.1007/s003550100139>.
- [40] J. K. Goeree, A. I. Kushnir, A Geometric Approach to Mechanism Design, *Social Science Research Network Working Paper Series* secondoftwo doi:, URL <http://dx.doi.org/10.2139/ssrn.1974922>.
- [41] M. Öztürk, H. Peters, T. Storcken, Strategy-proof location of a public bad on a disc, *Economics Letters* 119 (1) (2013) 14–16, ISSN 01651765, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.econlet.2013.01.012>.

- [42] M. Öztürk, H. Peters, T. Storcken, On the location of public bads: strategy-proofness under two-dimensional single-dipped preferences, *Economic Theory* 56 (1) (2014) 83–108, secondoftwo doi:, URL <http://dx.doi.org/10.1007/s00199-013-0785-8>.
- [43] K. Abe, A geometric approach to temptation, *Journal of Mathematical Economics* 48 (2) (2012) 92–97, ISSN 03044068, secondoftwo doi:, URL <http://dx.doi.org/10.1016/j.jmateco.2012.01.002>.
- [44] S. Michaelides, *Euclidean Economics: A Mathematical Method for Construction and Application of Economics Models*, Mesogios Publications - Ellinika Grammata, ISBN 9963-9176-0-7, 2006.
- [45] W. Thomson, Children Crying at Birthday Parties. Why?, *Economic Theory* 31 (3) (2007) 501–521, ISSN 0938-2259, secondoftwo doi:, URL <http://dx.doi.org/10.1007/s00199-006-0109-3>.
- [46] S. J. Brams, M. Jones, C. Klamler, Proportional pie-cutting, *International Journal of Game Theory* 36 (3-4) (2008) 353–367, secondoftwo doi:, URL http://www.nyu.edu/gsas/dept/politics/faculty/brams/prop_pie.pdf.
- [47] J. B. Barbanel, S. J. Brams, W. Stromquist, Cutting a pie is not a piece of cake, *American Mathematical Monthly* 116 (6) (2009) 496–514, secondoftwo doi:, URL http://mp.ra.uni-muenchen.de/12772/1/Pie-Cutting_rev_pdf.
- [48] T. P. Hill, Determining a Fair Border, *The American Mathematical Monthly* 90 (7) (1983) 438–442, ISSN 00029890, secondoftwo doi:, URL <http://dx.doi.org/10.2307/2975720>.
- [49] A. Beck, Constructing a Fair Border, *The American Mathematical Monthly* 94 (2) (1987) 157–162, ISSN 00029890, secondoftwo doi:, URL <http://dx.doi.org/10.2307/2322417>.
- [50] K. Iyer, M. N. Huhns, A Procedure for the Allocation of Two-Dimensional Resources in a Multiagent System, *International Journal of Cooperative Information Systems* 18 (2009) 1–34, secondoftwo doi:, URL <http://www.worldscientific.com/doi/abs/10.1142/S0218843009002051>.
- [51] T. Ichiishi, A. Idzik, Equitable allocation of divisible goods, *Journal of Mathematical Economics* 32 (4) (1999) 389–400, ISSN 03044068, secondoftwo doi:, URL [http://dx.doi.org/10.1016/s0304-4068\(98\)00053-6](http://dx.doi.org/10.1016/s0304-4068(98)00053-6).
- [52] R. Devulapalli, Geometric partitioning algorithms for fair division of geographic resources, Ph.D. thesis, University of Minnesota, URL <http://conservancy.umn.edu/handle/11299/165305>, 2014.
- [53] F. Maccheroni, M. Marinacci, How to cut a pizza fairly: Fair division with decreasing marginal evaluations, *Social Choice and Welfare* 20 (3) (2003) 457–465, secondoftwo doi:, URL <http://dx.doi.org/10.1007/s003550200192>.
- [54] M. Dall’Aglio, F. Maccheroni, Fair Division without Additivity, *American Mathematical Monthly* (2005) 363–365secondoftwo doi:, URL <http://dialnet.unirioja.es/servlet/articulo?codigo=1140450>.

- [55] N. Sagara, M. Vlach, Equity and Efficiency in a Measure Space with Nonadditive Preferences: The Problem of Cake Division, *Journal of Economic Literature* 90 (2005) 1, URL <http://web.econ.keio.ac.jp/staff/maruyama/rcme/seminar/abst050523.pdf>.
- [56] F. Hüsseinov, N. Sagara, Existence of efficient envy-free allocations of a heterogeneous divisible commodity with nonadditive utilities, *Social Choice and Welfare* (2013) 1–18secondoftwo doi:, URL <http://dx.doi.org/10.1007/s00355-012-0714-y>.
- [57] I. Caragiannis, J. K. Lai, A. D. Procaccia, Towards more expressive cake cutting, in: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, AAAI Press, ISBN 978-1-57735-513-7, 127–132, secondoftwo doi:, URL <http://www.cs.cmu.edu/~{}/arielpro/papers/nonadd.ijcai11.pdf>, 2011.
- [58] R. S. Mirchandani, Superadditivity and Subadditivity in Fair Division, *Journal of Mathematics Research* 5 (3) (2013) 78–91, ISSN 1916-9809, secondoftwo doi:, URL <http://dx.doi.org/10.5539/jmr.v5n3p78>.
- [59] C. Culter, Dividing people to rooms, *Mathematics Stack Exchange*, URL <http://math.stackexchange.com/q/461675>, uRL:<http://math.stackexchange.com/q/461675> (version: 2013-08-07), 2013.
- [60] M. V. Marathe, H. Breu, H. B. Hunt, S. S. Ravi, D. J. Rosenkrantz, Simple heuristics for unit disk graphs, *Networks* 25 (2) (1995) 59–68, secondoftwo doi:, URL <http://dx.doi.org/10.1002/net.3230250205>.
- [61] E. Segal-Halevi, A. Hassidim, Y. Aumann, Envy-Free Cake-Cutting in Two Dimensions, in: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, 1021–1028, URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9656>, 2015.
- [62] van der Stappen, D. Halperin, M. H. Overmars, The complexity of the free space for a robot moving amidst fat obstacles, *Computational Geometry* 3 (6) (1993) 353–373, ISSN 09257721, secondoftwo doi:, URL [http://dx.doi.org/10.1016/0925-7721\(93\)90007-s](http://dx.doi.org/10.1016/0925-7721(93)90007-s).
- [63] O. A. Nielsen, *An introduction to integration and measure theory*, Wiley, ISBN 0471595187, URL <http://www.worldcat.org/isbn/0471595187>, 1997.